What is accessibility?

- Make content usable by as many people as possible
- About 15% of world population lives with some form of disability: 1 billion people
- In the US, 1 in 5 adults has a disability
- Disabilities could be temporary or situational
Why is accessibility important?

- People with disabilities deserve equal rights
- The ADA and Section 508
- Accessible interface is about good design and coding practice
- Good accessibility is good user experience
What are we doing?

- Better knowledge sharing
- Review products internally for compliance
- Working accessibility into new features
What we will cover today

• Background
• Automated test
• Keyboard test
• Screen reader test
• Color accessibility
Why accessibility testing?

- Accessibility is about the *experience* of all users.
- Testing is the only way to ensure the experience is accessible.
Functional test

<table>
<thead>
<tr>
<th>Specification</th>
<th>WCAG 2.0 Success Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td>Verify how well web content functions as WCAG 2.0 specified</td>
</tr>
</tbody>
</table>

Less subjective compared to usability testing
Anyone can do the test!
### Overview of WCAG 2.0

<table>
<thead>
<tr>
<th>Principles</th>
<th>Success Criteria</th>
<th>Level A</th>
<th>Level AA</th>
<th>Level AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Perceivable</td>
<td>1.1 Text Alternatives</td>
<td>1.1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.2 Time-based Media</td>
<td>1.2.1 – 1.2.3</td>
<td>1.2.4 – 1.2.5</td>
<td>1.2.6 – 1.2.9</td>
</tr>
<tr>
<td></td>
<td>1.3 Adaptable</td>
<td>1.3.1 – 1.3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4 Distinguishable</td>
<td>1.4.1 – 1.4.2</td>
<td>1.4.3 – 1.4.5</td>
<td>1.4.6 – 1.4.9</td>
</tr>
<tr>
<td>2. Operable</td>
<td>2.1 Keyboard Accessible</td>
<td>2.1.1 – 2.1.2</td>
<td></td>
<td>2.1.3</td>
</tr>
<tr>
<td></td>
<td>2.2 Enough Time</td>
<td>2.2.1 – 2.2.2</td>
<td>2.2.3 – 2.2.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.3 Seizures</td>
<td>2.3.1</td>
<td></td>
<td>2.3.2</td>
</tr>
<tr>
<td></td>
<td>2.4 Navigable</td>
<td>2.4.1 – 2.4.4</td>
<td>2.4.5 – 2.4.7</td>
<td>2.4.8 – 2.4.10</td>
</tr>
<tr>
<td>3. Understandable</td>
<td>3.1 Readable</td>
<td>3.1.1</td>
<td>3.1.2</td>
<td>3.1.3 – 3.1.6</td>
</tr>
<tr>
<td></td>
<td>3.2 Predictable</td>
<td>3.2.1 – 3.2.2</td>
<td>3.2.3 – 3.2.4</td>
<td>3.2.5</td>
</tr>
<tr>
<td></td>
<td>3.3 Input Assistance</td>
<td>3.3.1 – 3.3.2</td>
<td>3.3.3 – 3.3.4</td>
<td>3.3.5 – 3.3.6</td>
</tr>
<tr>
<td>4. Robust</td>
<td>4.1 Compatible</td>
<td>4.1.1 – 4.1.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Test process

Automated → Keyboard → Screen reader → Color
Automated test
- Tests rendered browser DOM
- Aims at no false positives
- Accessible
- Helpful documentation
Practice aXe

Park Locator

http://arcg.is/05DzDX
Automated test

• No automated test tools can definitely prove conformance with any given WCAG Success Criterion.
• Automated test is good starting point but cannot detect all accessibility issues.
• Run automated test of each page state.
Keyboard navigation

- tab
- shift
- tab

Move keyboard focus

- enter

Click links

- enter
- space

Click buttons

- ↑
- ↓
- ←
- →

Menus and some form controls
Expected outcomes

- 2.1.1: Interact with all controls, links, and menus using only keyboard.
- 2.4.7: See what item has focus at all times.
- 2.4.3: Visual focus order matches intended interaction order.
- 2.1.2: No keyboard trap.
- Off-screen content (e.g., responsive navigation) should not receive focus when invisible.
Practice keyboard test

Test cases
- Park Locator
- UK Government Digital Service – Test cases (https://arcg.is/1OLC8K)
- Accessible University (http://arcg.is/1O45jz)

Tools
- Enhanced focus (http://arcg.is/19muKy)
- document.activeElement
Screen reader test
## Screen reader

Recommended combinations:

<table>
<thead>
<tr>
<th>OS</th>
<th>Screen reader</th>
<th>Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacOS</td>
<td>VoiceOver</td>
<td>Safari</td>
</tr>
<tr>
<td>Windows</td>
<td>NVDA</td>
<td>Firefox</td>
</tr>
<tr>
<td>Windows</td>
<td>JAWS</td>
<td>IE/Edge</td>
</tr>
</tbody>
</table>
## Screen reader

<table>
<thead>
<tr>
<th></th>
<th>Turn on</th>
<th>Stop</th>
<th>Modifier key</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VoiceOver</strong></td>
<td>Command + F5</td>
<td>Command + F5</td>
<td>Control + Option</td>
</tr>
<tr>
<td><strong>NVDA</strong></td>
<td>Control + Alt + N</td>
<td>NVDA + Q</td>
<td>Numpad Insert</td>
</tr>
<tr>
<td><strong>JAWS</strong></td>
<td>Control + Alt + J</td>
<td>Insert + F4</td>
<td>Numpad Insert</td>
</tr>
</tbody>
</table>

Modifier key: Enter screen reader commands by pressing modifier key and one or more other keys.
Screen reader commands

- VoiceOver
- NVDA
- JAWS
- Basic commands for accessibility testing
<table>
<thead>
<tr>
<th>VoiceOver commands</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>VO + right/left arrow</td>
<td>Read next/previous item</td>
</tr>
<tr>
<td>Control</td>
<td>Stop reading</td>
</tr>
<tr>
<td>VO + space</td>
<td>Click link, button, form controls</td>
</tr>
<tr>
<td>VO + u</td>
<td>Open rotor</td>
</tr>
</tbody>
</table>
Screen reader testing coverage

<table>
<thead>
<tr>
<th>Navigation</th>
<th>Content</th>
<th>Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headings</td>
<td>Alt text</td>
<td>Forms</td>
</tr>
<tr>
<td>Links</td>
<td>Tables</td>
<td>Dialogs</td>
</tr>
<tr>
<td>Landmarks</td>
<td>Charts</td>
<td>Messages</td>
</tr>
<tr>
<td>Menus</td>
<td></td>
<td>Widgets</td>
</tr>
</tbody>
</table>
WAI-ARIA Authoring Practices

TABLE OF CONTENTS

1. Introduction
2. Read Me First
2.1 No ARIA is better than Bad ARIA
2.2 Browser and Assistive Technology Support
2.3 Mobile and Touch Support
3. Design Patterns and Widgets
3.1 Accordion (Sections With Show/Hide Functionality)
3.2 Alert
3.3 Alert and Message Dialogs
3.4 Breadcrumb
3.5 Button
3.6 Checkbox
3.7 Combo Box
3.8 Dialog (Modal)
3.9 Disclosure (Show/Hide)
3.10 Feed
3.11 Grids : Interactive Tabular Data and Layout Containers
3.12 Link
3.13 Listbox
3.14 Menu or Menu bar
3.15 Menu Button
3.16 Radio Group

WAI-ARIA Authoring Practices 1.1
W3C Working Group Note 14 December 2017

This version:
https://www.w3.org/TR/2017/NOTE-wai-aria-practices-1.1-20171214/

Latest published version:
https://www.w3.org/TR/wai-aria-practices-1.1/

Latest editor's draft:
https://w3c.github.io/aria-practices/

Previous version:
https://www.w3.org/TR/2017/WD-wai-aria-practices-1.1-20170628/

Editors:
Matt King, Facebook, mok@fb.com
James Nurthen, Oracle Corporation, james.nurthen@oracle.com
Michiel Blij, Invited Expert, Michael.Cooper@w3.org
Joseph Scheuhammer, Inclusive Design Research Centre, OCAD University (Previous Editor)
Lisa Pappas, SAS (Previous Editor)
Rich Schwartzfeger, IBM Corporation (Previous Editor)

Copyright © 2015-2017 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply.

Abstract

This document provides readers with an understanding of how to use WAI-ARIA 1.1 [wai-aria-1.1] to create accessible rich internet applications. It describes considerations that might not be evident to most authors from the WAI-ARIA specification alone and recommends approaches to make widgets, navigation, and behaviors accessible using WAI-ARIA roles, states, and properties. This document is directed primarily to Web application developers, but the guidance is also useful for user agent and assistive technology developers.

https://www.w3.org/TR/wai-aria-practices-1.1/
Practice screen reader

Park Locator

http://arcg.is/05DzDX
Color test
Expected outcomes

- 1.4.1: Not use presentation that relies solely on color.
- 1.4.3: Color contrast ratio is at least 4.5:1.
Practice color test

Test cases
- Park Locator

Tools
- Contrast ratio calculator (http://arcg.is/1m44TW)
Summary

• Start with automated test, then do keyboard, screen reader, and color test.
• Need to understand WCAG 2.0 Success Criteria.
• Get familiar with ARIA for widgets.
The ultimate decision-maker about whether or not something is accessible, is whether or not people can use it.
Want to learn more about accessibility?

Accessible Web Mapping Apps: ARIA, WCAG and 508 Compliance
Kelly Hutchins, Tao Zhang

8:30 – 9:30 am
Friday, March 9
Smoketree A-E
Schedule

Hands-on Workshops

Tuesday, March 6
1:00 – 2:00 PM
Introducing UX to your GIS Org
Mesquite B

4:00 – 5:00 PM
DIY Usability Testing
Mesquite C

Thursday, March 8
9:00 – 10:00 AM
DIY Accessibility
Mesquite B

1:00 – 2:00 PM
ArcGIS Runtime: Building Great User Experience
Smoketree A-E

2:30 – 3:30 PM
Collaborative Brainstorming
Mesquite G-H

4:00 – 5:00 PM
Customizing the ArcGIS API for JavaScript Widgets
Primrose A

Friday, March 9
8:30 – 9:30 AM
Accessible Web Mapping Apps

10:00 – 11:00 AM
DIY Usability Testing
Smoketree A-E
Please Take Our Survey!

Download the Esri Events app and find your event

Select the session you attended

Scroll down to the “Feedback” section

Complete Answers, add a Comment, and Select “Submit”