Accessible Web Mapping Apps

Kelly Hutchins

Tao Zhang

What is accessibility?

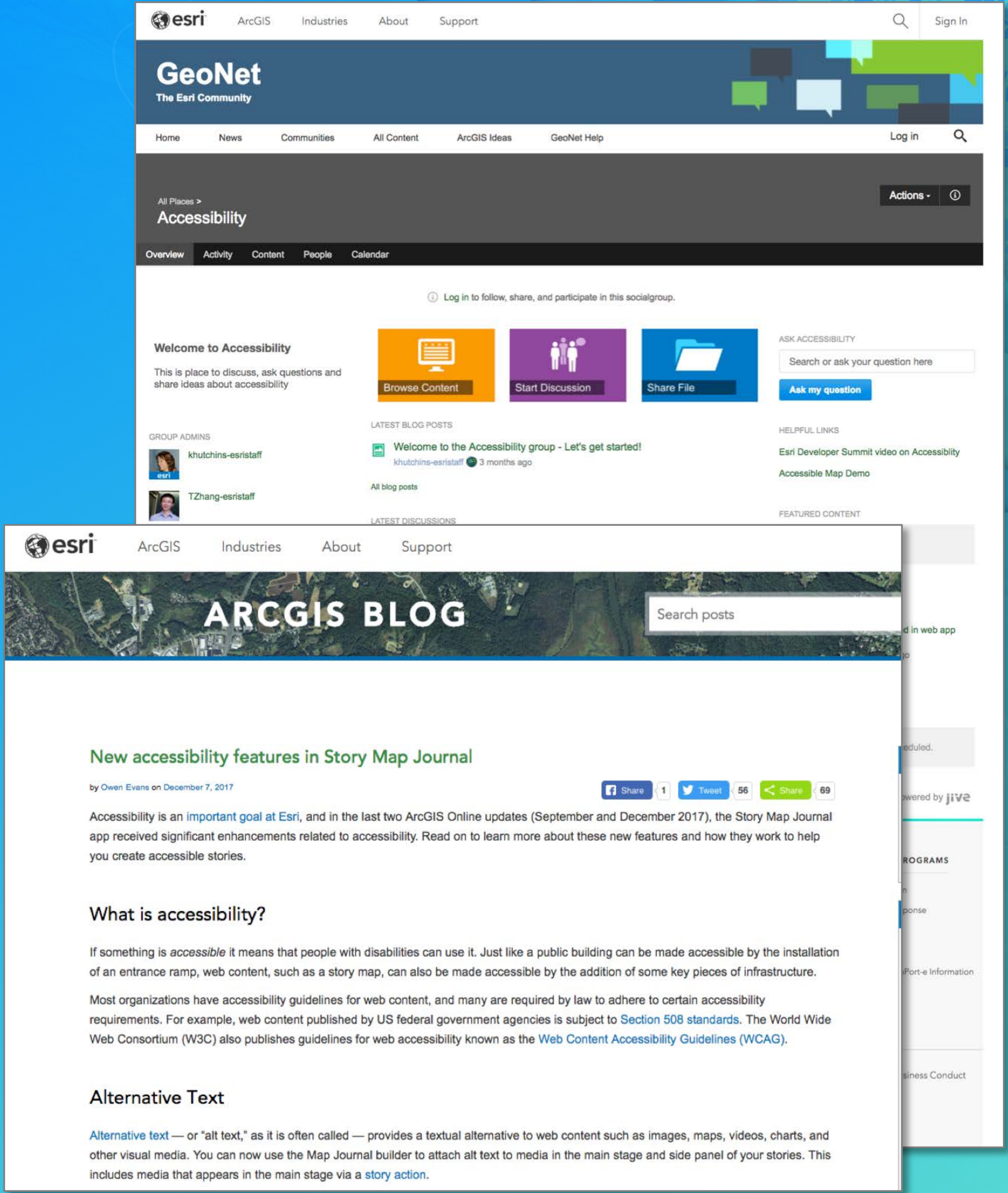
- Make content usable by as many people as possible
- About 15% of world population lives with some form of disability: **1 billion** people
- In the US, 1 in 5 adults has a disability
- Disabilities could be temporary or situational

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



Log in to follow, share, and participate in this socialgroup.

Welcome to Accessibility

This is place to discuss, ask questions and share ideas about accessibility

Welcome to the Accessibility group - Let's get started!
khutchins-esristaff 3 months ago

New accessibility features in Story Map Journal

by Owen Evans on December 7, 2017

Accessibility is an important goal at Esri, and in the last two ArcGIS Online updates (September and December 2017), the Story Map Journal app received significant enhancements related to accessibility. Read on to learn more about these new features and how they work to help you create accessible stories.

What is accessibility?

If something is *accessible* it means that people with disabilities can use it. Just like a public building can be made accessible by the installation of an entrance ramp, web content, such as a story map, can also be made accessible by the addition of some key pieces of infrastructure.

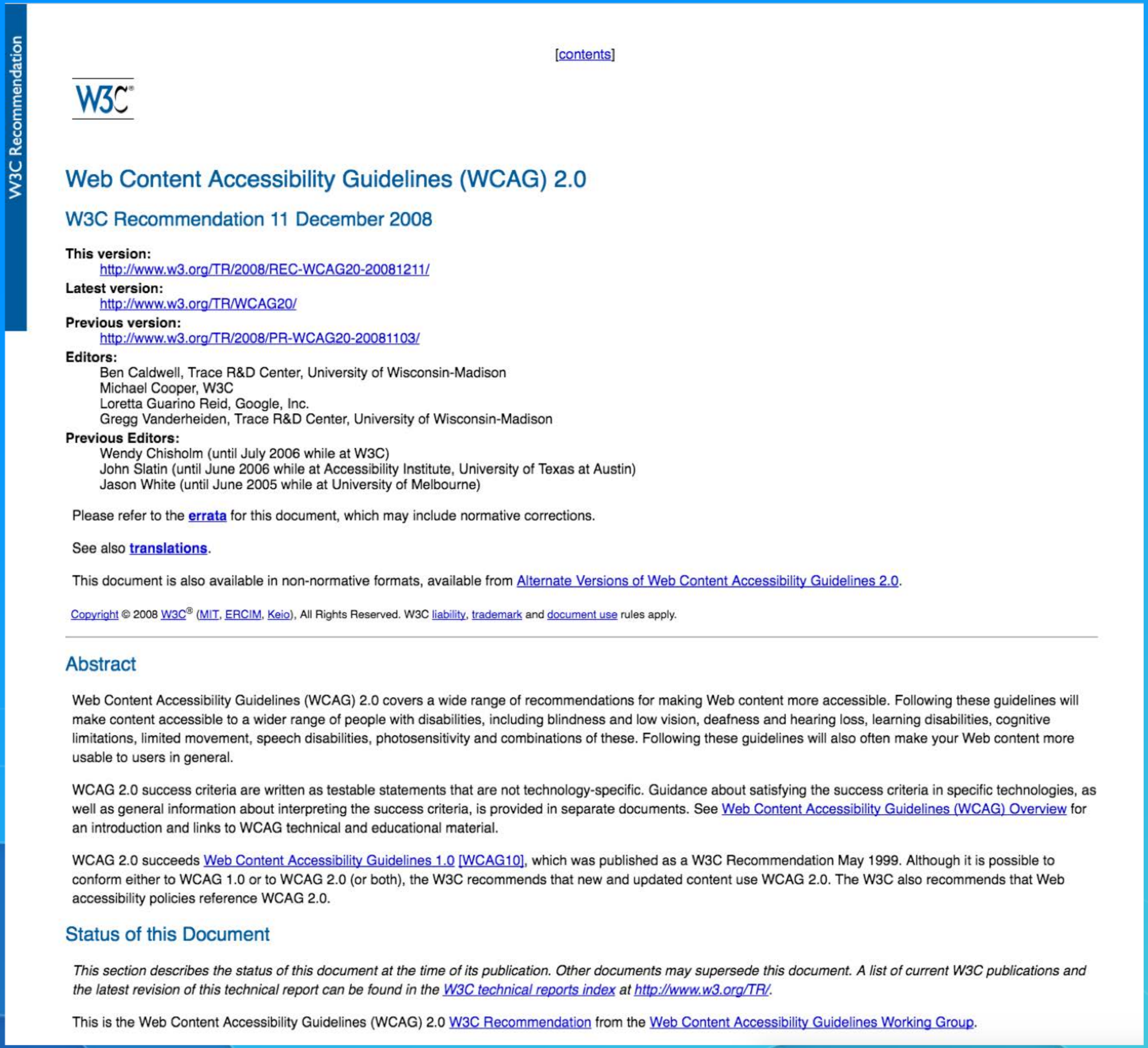
Most organizations have accessibility guidelines for web content, and many are required by law to adhere to certain accessibility requirements. For example, web content published by US federal government agencies is subject to Section 508 standards. The World Wide Web Consortium (W3C) also publishes guidelines for web accessibility known as the Web Content Accessibility Guidelines (WCAG).

Alternative Text

Alternative text — or "alt text," as it is often called — provides a textual alternative to web content such as images, maps, videos, charts, and other visual media. You can now use the Map Journal builder to attach alt text to media in the main stage and side panel of your stories. This includes media that appears in the main stage via a story action.

Web Content Accessibility Guidelines

WCAG



[contents]

W3C®

Web Content Accessibility Guidelines (WCAG) 2.0

W3C Recommendation 11 December 2008

This version:
http://www.w3.org/TR/2008/REC-WCAG20-20081211/

Latest version:
http://www.w3.org/TR/WCAG20/

Previous version:
http://www.w3.org/TR/2008/PR-WCAG20-20081103/

Editors:
Ben Caldwell, Trace R&D Center, University of Wisconsin-Madison
Michael Cooper, W3C
Loretta Guarino Reid, Google, Inc.
Gregg Vanderheiden, Trace R&D Center, University of Wisconsin-Madison

Previous Editors:
Wendy Chisholm (until July 2006 while at W3C)
John Slatin (until June 2006 while at Accessibility Institute, University of Texas at Austin)
Jason White (until June 2005 while at University of Melbourne)

Please refer to the **errata** for this document, which may include normative corrections.

See also **translations**.

This document is also available in non-normative formats, available from Alternate Versions of Web Content Accessibility Guidelines 2.0.

Abstract

Web Content Accessibility Guidelines (WCAG) 2.0 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content accessible to a wider range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity and combinations of these. Following these guidelines will also often make your Web content more usable to users in general.

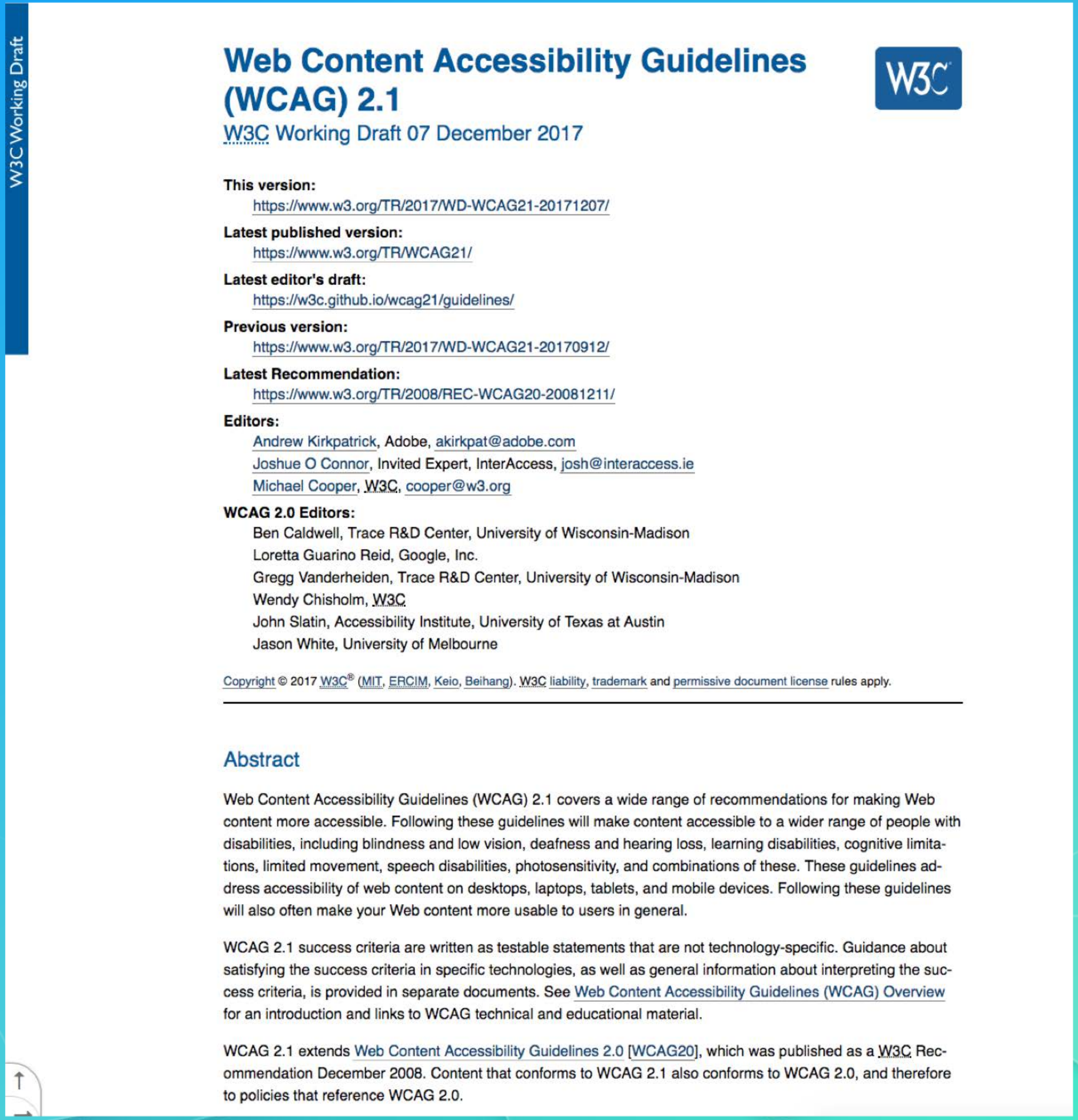
WCAG 2.0 success criteria are written as testable statements that are not technology-specific. Guidance about satisfying the success criteria in specific technologies, as well as general information about interpreting the success criteria, is provided in separate documents. See Web Content Accessibility Guidelines (WCAG) Overview for an introduction and links to WCAG technical and educational material.

WCAG 2.0 succeeds Web Content Accessibility Guidelines 1.0 [WCAG10], which was published as a W3C Recommendation May 1999. Although it is possible to conform either to WCAG 1.0 or to WCAG 2.0 (or both), the W3C recommends that new and updated content use WCAG 2.0. The W3C also recommends that Web accessibility policies reference WCAG 2.0.

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This is the Web Content Accessibility Guidelines (WCAG) 2.0 W3C Recommendation from the Web Content Accessibility Guidelines Working Group.



Web Content Accessibility Guidelines (WCAG) 2.1

W3C®

W3C Working Draft 07 December 2017

This version:
https://www.w3.org/TR/2017/WD-WCAG21-20171207/

Latest published version:
https://www.w3.org/TR/WCAG21/

Latest editor's draft:
https://w3c.github.io/wcag21/guidelines/

Previous version:
https://www.w3.org/TR/2017/WD-WCAG21-20170912/

Latest Recommendation:
https://www.w3.org/TR/2008/REC-WCAG20-20081211/

Editors:
Andrew Kirkpatrick, Adobe, akirkpat@adobe.com
Joshue O Connor, Invited Expert, InterAccess, josh@interaccess.ie
Michael Cooper, W3C, cooper@w3.org

WCAG 2.0 Editors:
Ben Caldwell, Trace R&D Center, University of Wisconsin-Madison
Loretta Guarino Reid, Google, Inc.
Gregg Vanderheiden, Trace R&D Center, University of Wisconsin-Madison
Wendy Chisholm, W3C
John Slatin, Accessibility Institute, University of Texas at Austin
Jason White, University of Melbourne

Abstract

Web Content Accessibility Guidelines (WCAG) 2.1 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content accessible to a wider range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity, and combinations of these. These guidelines address accessibility of web content on desktops, laptops, tablets, and mobile devices. Following these guidelines will also often make your Web content more usable to users in general.

WCAG 2.1 success criteria are written as testable statements that are not technology-specific. Guidance about satisfying the success criteria in specific technologies, as well as general information about interpreting the success criteria, is provided in separate documents. See Web Content Accessibility Guidelines (WCAG) Overview for an introduction and links to WCAG technical and educational material.

WCAG 2.1 extends Web Content Accessibility Guidelines 2.0 [WCAG20], which was published as a W3C Recommendation December 2008. Content that conforms to WCAG 2.1 also conforms to WCAG 2.0, and therefore to policies that reference WCAG 2.0.

Overview of WCAG 2.0

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

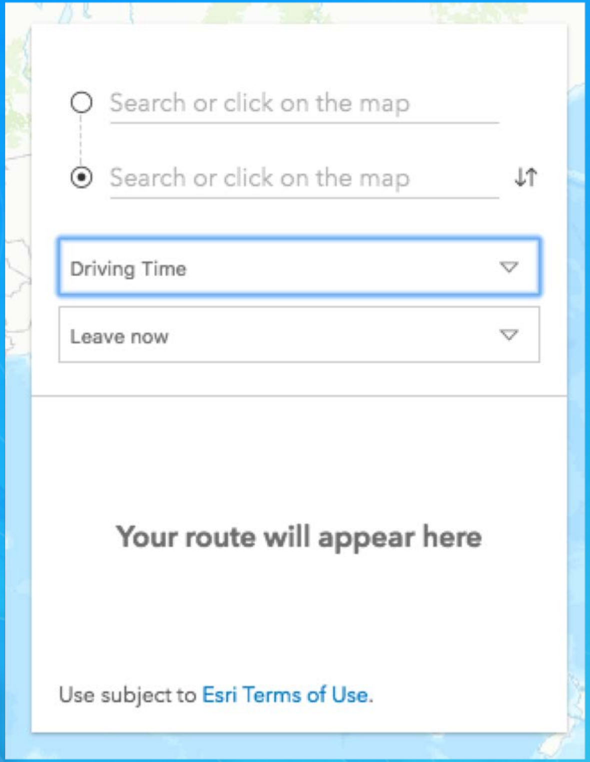
What we will cover today

- Focus and keyboard
- Color
- Information and relationships
- Testing
- Accessible map

Focus and keyboard

Focus

- WCAG 2.4.7: Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.

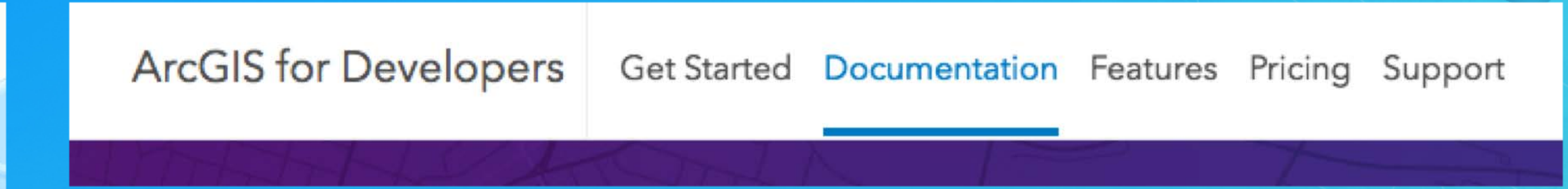


Driving Time

Leave now

Your route will appear here

Use subject to Esri Terms of Use.

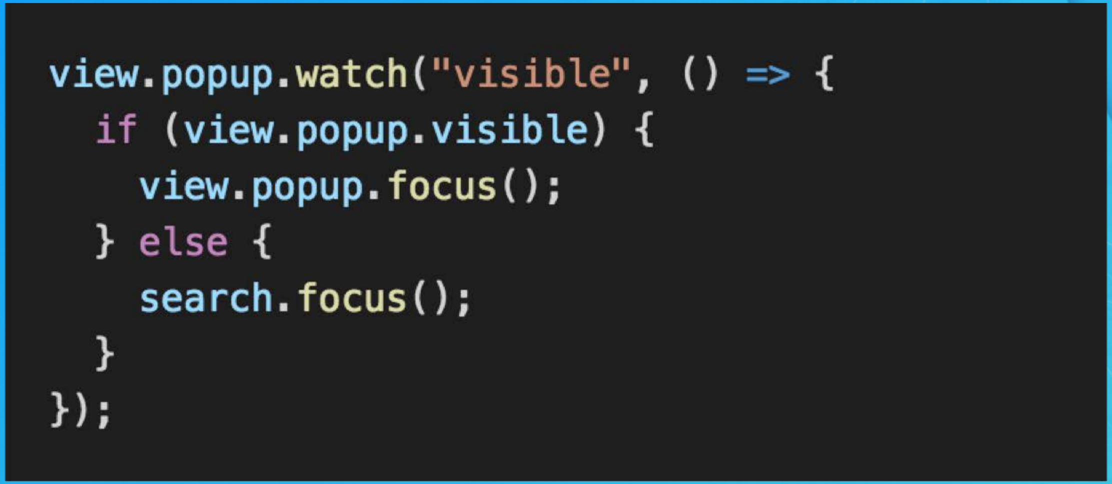


- Interactive HTML elements should have clear focus
- Don't set `outline: 0px` for `:focus`
- For links use `text-decoration: underline` for `:hover &` `:focus`, or make it more decorative

ArcGIS API for JavaScript

Classes with `focus` method:

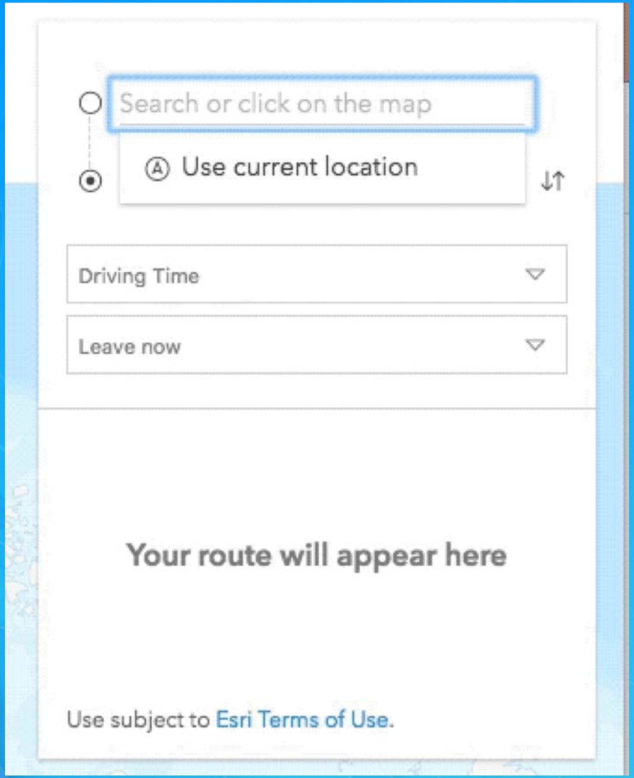
- `MapView` and `SceneView`
- `Popup`
- `Search`



```
view.popup.watch("visible", () => {
  if (view.popup.visible) {
    view.popup.focus();
  } else {
    search.focus();
  }
});
```

Tab order

- WCAG 1.3.2: Navigation order, as determined by DOM structure, should be logical and intuitive.



○ Search or click on the map

Ⓐ Use current location

Driving Time

Leave now

Your route will appear here

Use subject to Esri Terms of Use.

- Be careful changing visual position of elements on screen using CSS
- Avoid jumping around tab order

Tab order

`tabindex=“0”`	Let natural DOM structure determine tab order
`tabindex=“-1”`	If need to programmatically move focus by calling `focus()`
`tabindex=“4”`	Anti-pattern

Keyboard

- WCAG 2.1.1: Keyboard users should be able to use all page functionalities using keyboard only.
- Native interactive elements like `<a>`, `<button>`, and `<input>` receive keyboard actions
- Support appropriate keystrokes as if it were native element (e.g., Space & Enter on button)

Map navigation

Action	Behavior
Arrow keys	Nudge the view to the left, right, up, or down
N	Adjust the view to point north
A	Rotate the view counter clockwise
D	Rotate the view clockwise
+	Incrementally zoom in
-	Incrementally zoom out

https://developers.arcgis.com/javascript/latest/api-reference/esri-views-MapView.html#navigation

SceneView navigation

Action	Behavior
Arrow keys	Nudge the scene to the left, right, up, or down
P	Move the camera to look perpendicular
N	Adjust the scene to point north
J	Move down closer to the view
U	Move up, higher from the view

https://developers.arcgis.com/javascript/latest/api-reference/esri-views-SceneView.html#navigation

Keyboard example

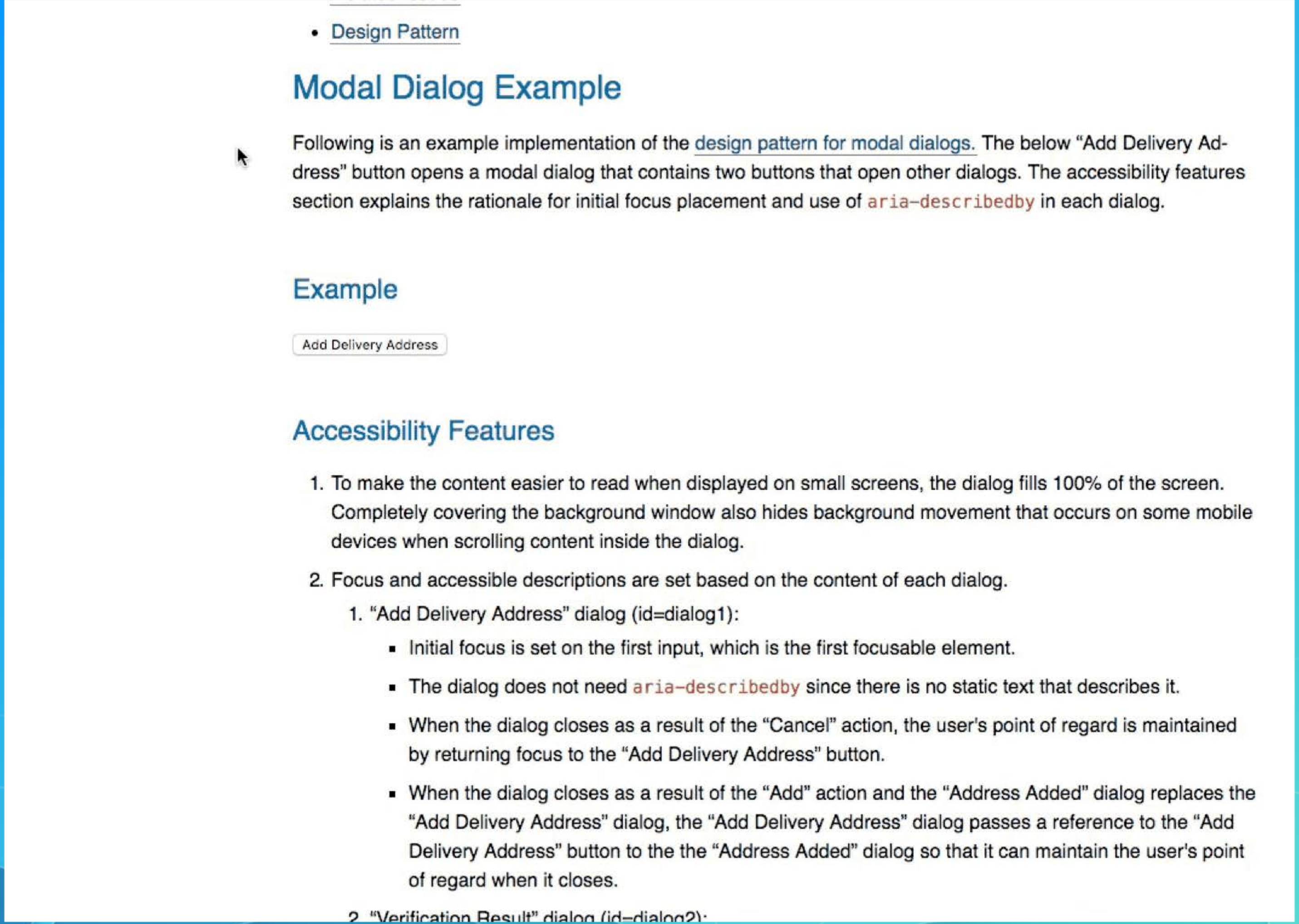
```
<div class="button dark” role="button” tabindex="0” id="buttonId">
     Search
</div>
```

```
document.getElementById("buttonId")
    .addEventListener("keyup",function(event) {
        event.preventDefault();
        if((event.keyCode === 13)||(event.keyCode === 32)) {
            // respond to Enter/Space keys;
        }
});
```

No keyboard trap

- WCAG 2.1.2: Content does not “trap” keyboard focus within subsections.
- Keyboard focus stay inside modal dialog until dialog dismissed.
- Restore keyboard focus to previously focused element after dialog dismissed.

Keyboard trap example



- Design Pattern

Modal Dialog Example

Following is an example implementation of the design pattern for modal dialogs. The below "Add Delivery Address" button opens a modal dialog that contains two buttons that open other dialogs. The accessibility features section explains the rationale for initial focus placement and use of `aria-describedby` in each dialog.

Example

Accessibility Features

1. To make the content easier to read when displayed on small screens, the dialog fills 100% of the screen. Completely covering the background window also hides background movement that occurs on some mobile devices when scrolling content inside the dialog.
2. Focus and accessible descriptions are set based on the content of each dialog.
   1. "Add Delivery Address" dialog (id=dialog1):
      - Initial focus is set on the first input, which is the first focusable element.
      - The dialog does not need `aria-describedby` since there is no static text that describes it.
      - When the dialog closes as a result of the "Cancel" action, the user's point of regard is maintained by returning focus to the "Add Delivery Address" button.
      - When the dialog closes as a result of the "Add" action and the "Address Added" dialog replaces the "Add Delivery Address" dialog, the "Add Delivery Address" dialog passes a reference to the "Add Delivery Address" button to the the "Address Added" dialog so that it can maintain the user's point of regard when it closes.
   2. "Verification Result" dialog (id=dialog2):

https://www.w3.org/TR/wai-aria-practices/examples/dialog-modal/dialog.html

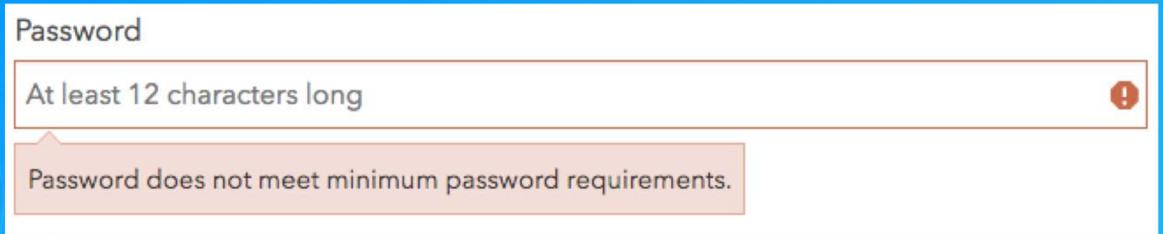
Test focus and keyboard

- Tab through page to verify all interactive elements have focus
- Visual focus order matches intended interaction order
- Interact with all controls, links, and menus using only keyboard
- No keyboard trap except for modals
- Off-screen content like responsive navigation should not receive focus when invisible

Color contrast

Use of color

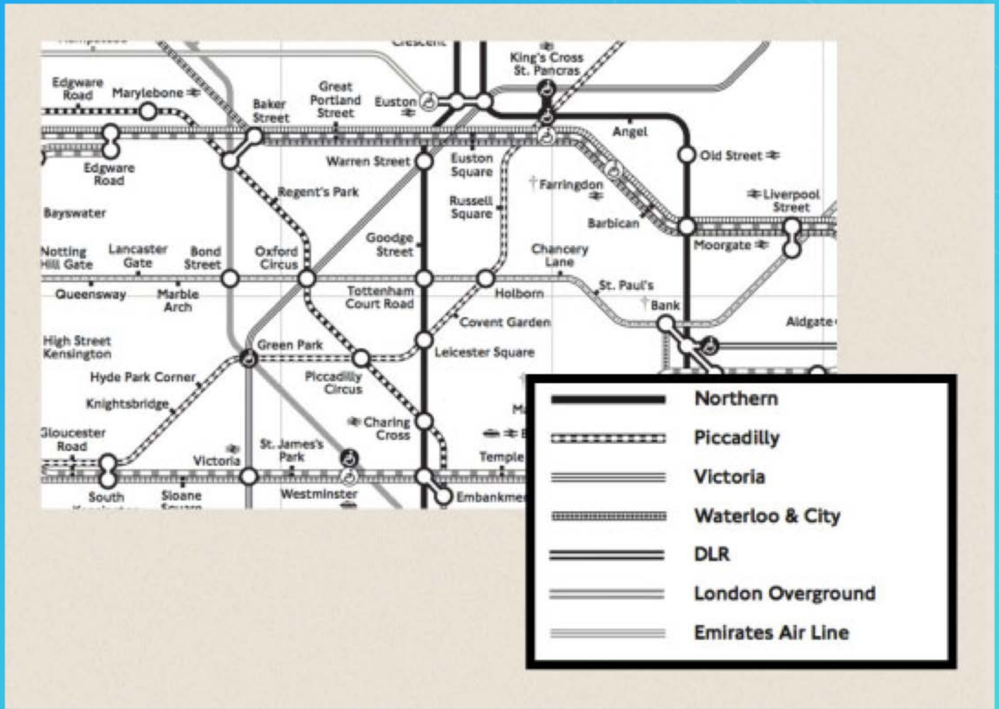
- WCAG 1.4.1: Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.



Password

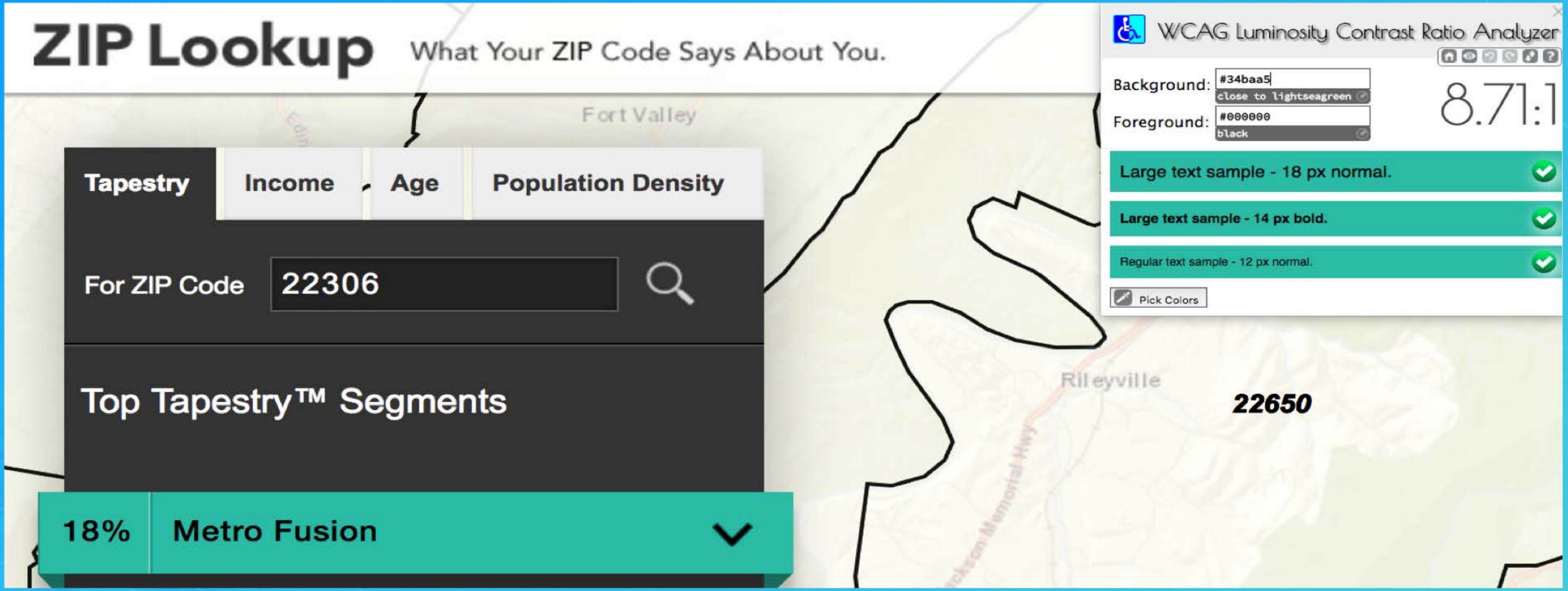
At least 12 characters long

Password does not meet minimum password requirements.



Minimum contrast

- WCAG 1.4.3: The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for large text (ratio of at least 3:1), incidental text, or logotype.



ZIP Lookup What Your ZIP Code Says About You.

For ZIP Code 22306

Top Tapestry™ Segments

18% Metro Fusion

WCAG Luminosity Contrast Ratio Analyzer

Background: #34baa5 close to lightseagreen

Foreground: #000000 black

8.71:1

Large text sample - 18 px normal. ✓

Large text sample - 14 px bold. ✓

Regular text sample - 12 px normal. ✓

Test color

- Look for content differentiated by color only
- Run an automated test tool: aXe
- Use a contrast ratio calculator

Info and relationships

Clear semantics

- WCAG 1.3.1: Content should have good semantic structure.
- Assistive Technologies (AT) rely on semantic code to drive their behavior.
- Information architecture is important.
- Non-sighted users should be able to infer the same information as sighted users without depending on colors, shape, or typography for context.

Semantics example

```
<li tabindex="0” class="checkbox” checked>
    Show premium content
</li>
```

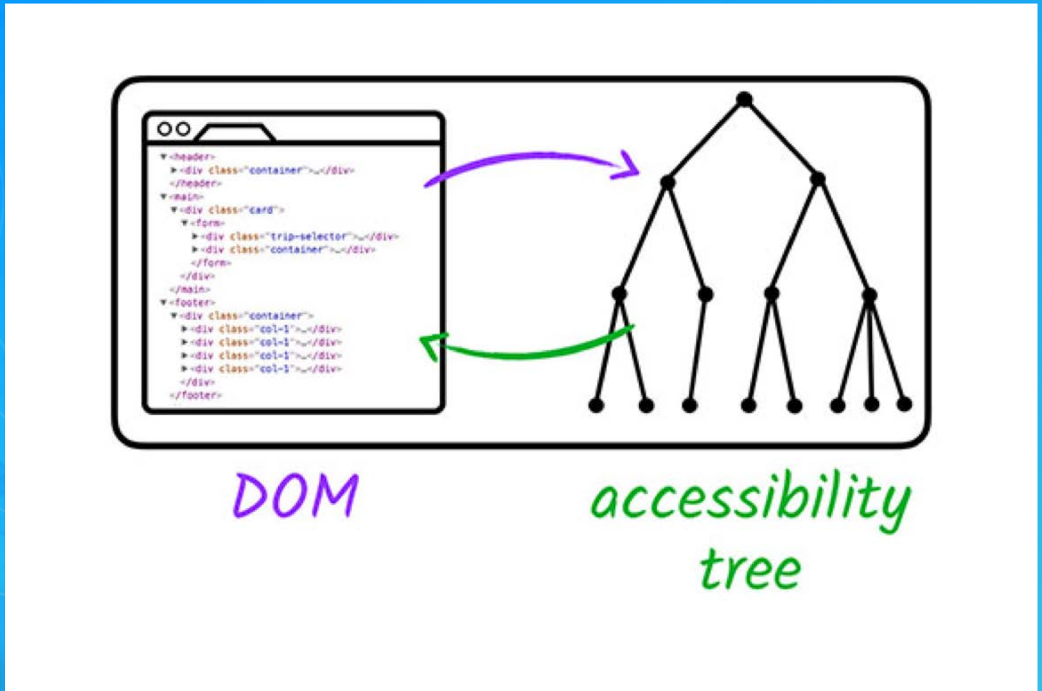
- Sighted users see a checkbox as a result of CSS class `checkbox`
- AT users will not know this is meant to be a checkbox

```
<li tabindex="0” class="checkbox”
role="checkbox” checked aria-checked="true">
    Show premium content
</li>
```

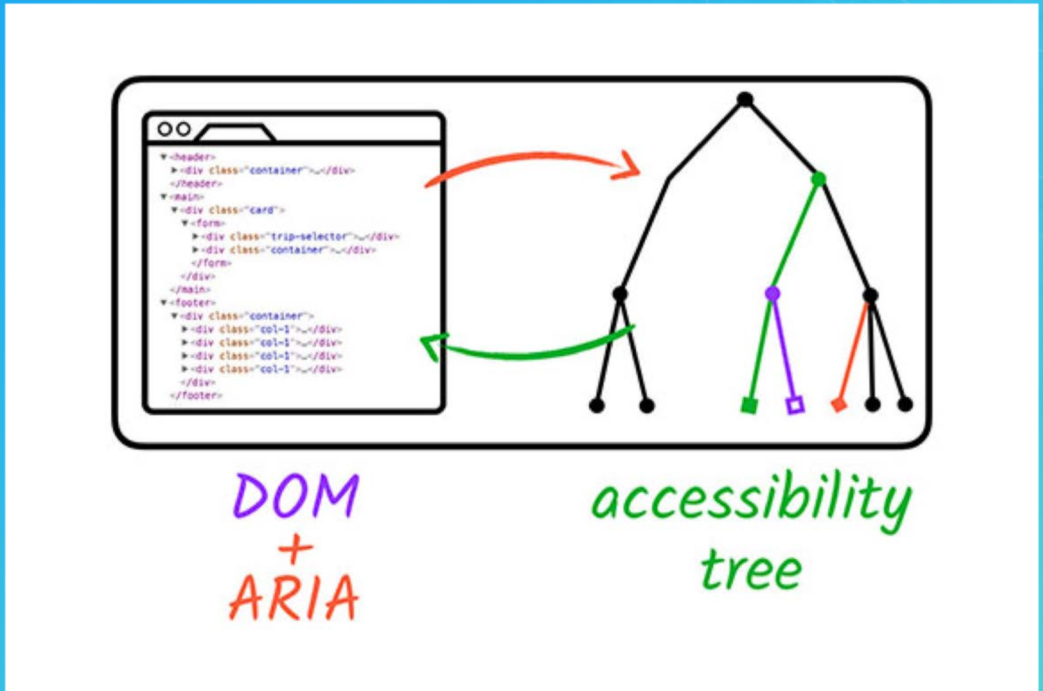
- The `checkbox` role is used for checkable interactive controls.
- It’s also required to have `aria-checked` attribute to expose checkbox state to AT users.

WAI-ARIA

- Specification for increasing accessibility of custom elements.
- Allows developers to modify and augment standard DOM to accessibility tree for AT.



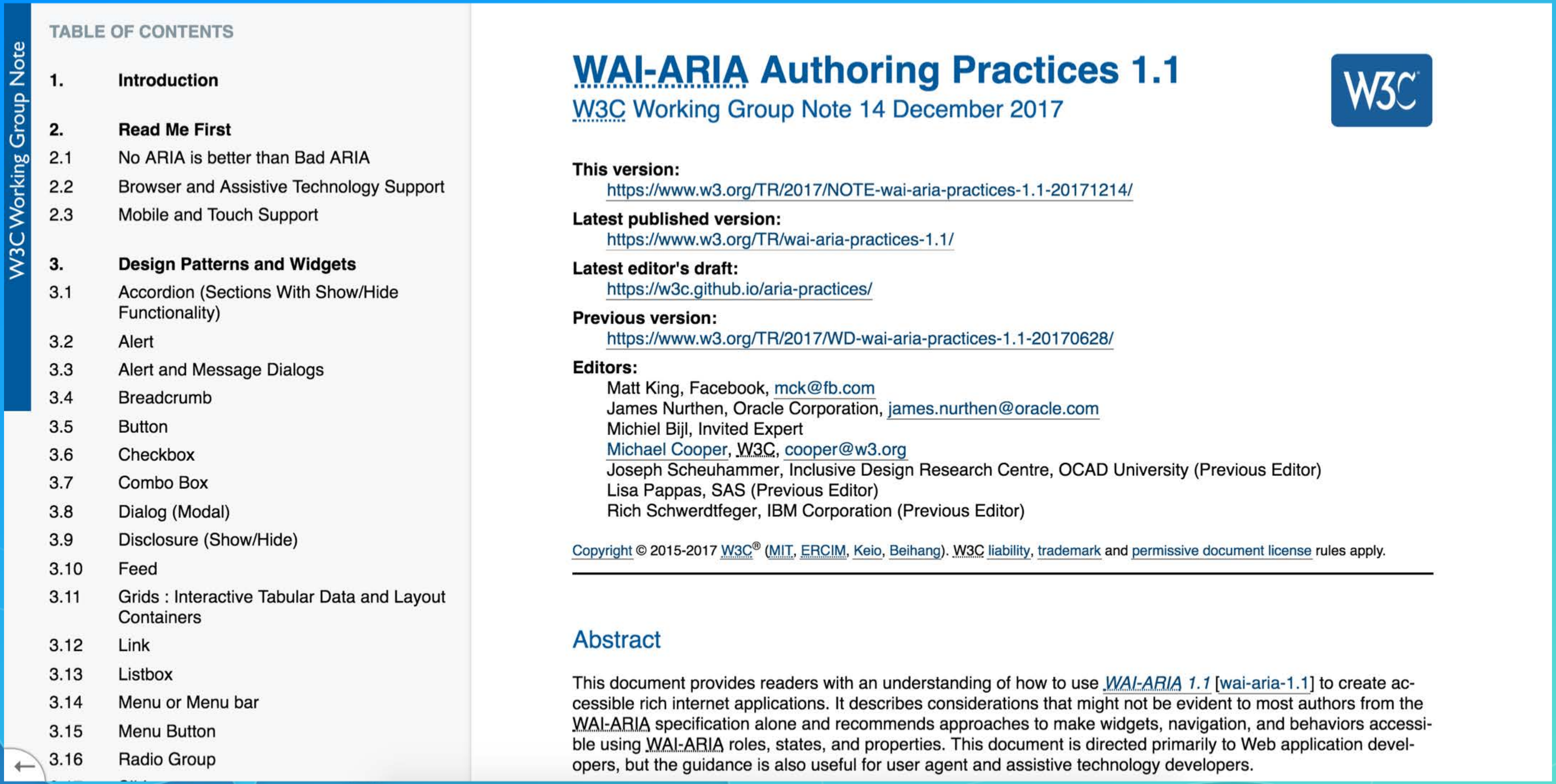
```
▼<header>
  ▶<div class="container">…</div>
 </header>
▼<main>
  ▼<div class="card">
    ▼<form>
      ▶<div class="trip-selector">…</div>
      ▶<div class="container">…</div>
     </form>
    </div>
 </main>
▼<footer>
  ▼<div class="container">
    ▶<div class="col-1">…</div>
    ▶<div class="col-1">…</div>
    ▶<div class="col-1">…</div>
    ▶<div class="col-1">…</div>
    </div>
 </footer>
```



```
▼<header>
  ▶<div class="container">…</div>
 </header>
▼<main>
  ▼<div class="card">
    ▼<form>
      ▶<div class="trip-selector">…</div>
      ▶<div class="container">…</div>
     </form>
    </div>
 </main>
▼<footer>
  ▼<div class="container">
    ▶<div class="col-1">…</div>
    ▶<div class="col-1">…</div>
    ▶<div class="col-1">…</div>
    ▶<div class="col-1">…</div>
    </div>
 </footer>
```

https://developers.google.com/web/fundamentals/accessibility/semantics-aria/

WAI-ARIA Authoring Practices



W3C Working Group Note

TABLE OF CONTENTS

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, mck@fb.com
James Nurthen, Oracle Corporation, james.nurthen@oracle.com
Michiel Bijl, Invited Expert
Michael Cooper, W3C, cooper@w3.org
Joseph Scheuhammer, Inclusive Design Research Centre, OCAD University (Previous Editor)
Lisa Pappas, SAS (Previous Editor)
Rich Schwerdtfeger, IBM Corporation (Previous Editor)

Abstract

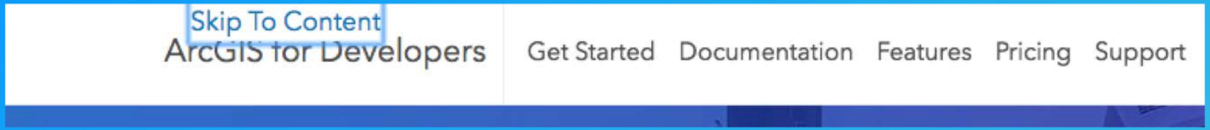
This document provides readers with an understanding of how to use *WAI-ARIA 1.1* [wai-aria-1.1] to create accessible rich internet applications. It describes considerations that might not be evident to most authors from the WAI-ARIA specification alone and recommends approaches to make widgets, navigation, 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.

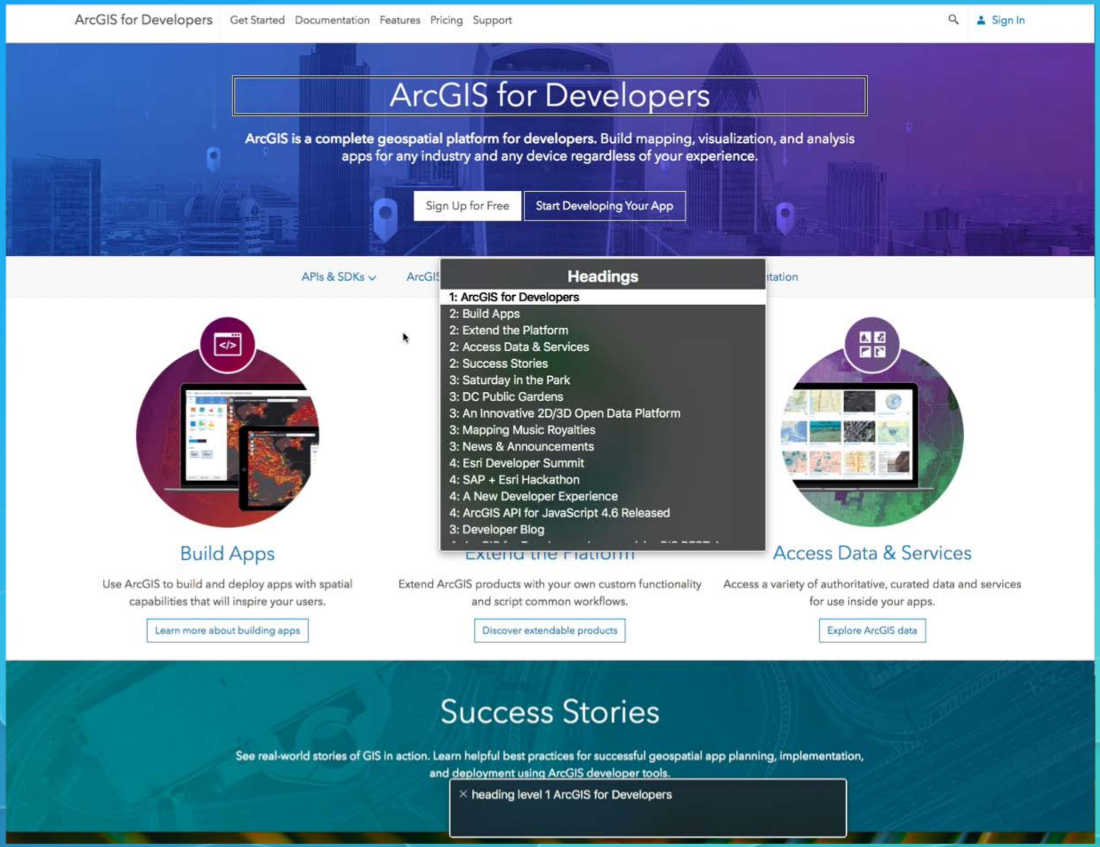
https://www.w3.org/TR/wai-aria-practices-1.1/

Bypass blocks

- WCAG 2.4.1: Pages should have mechanisms like Skip-Navigation and WAI ARIA landmark roles for users to jump to main content or a particular content area.

Bypass blocks example





ArcGIS for Developers

ArcGIS is a complete geospatial platform for developers. Build mapping, visualization, and analysis apps for any industry and any device regardless of your experience.

Headings

1: ArcGIS for Developers
2: Build Apps
2: Extend the Platform
2: Access Data & Services
2: Success Stories
3: Saturday in the Park
3: DC Public Gardens
3: An Innovative 2D/3D Open Data Platform
3: Mapping Music Royalties
3: News & Announcements
4: Esri Developer Summit
4: SAP + Esri Hackathon
4: A New Developer Experience
4: ArcGIS API for JavaScript 4.6 Released
3: Developer Blog

Build Apps

Use ArcGIS to build and deploy apps with spatial capabilities that will inspire your users.

Extend the Platform

Extend ArcGIS products with your own custom functionality and script common workflows.

Access Data & Services

Access a variety of authoritative, curated data and services for use inside your apps.

Success Stories

See real-world stories of GIS in action. Learn helpful best practices for successful geospatial app planning, implementation, and deployment using ArcGIS developer tools.

× heading level 1 ArcGIS for Developers

Labels or instructions

- WCAG 3.3.2: Labels or instructions are provided when content requires user input.
- Labels should be associated with form inputs.

Label examples

Associate label implicitly

```
<label>
    Email
    <input type="text” placeholder="name@example.com">
</label>
```

Associate label explicitly

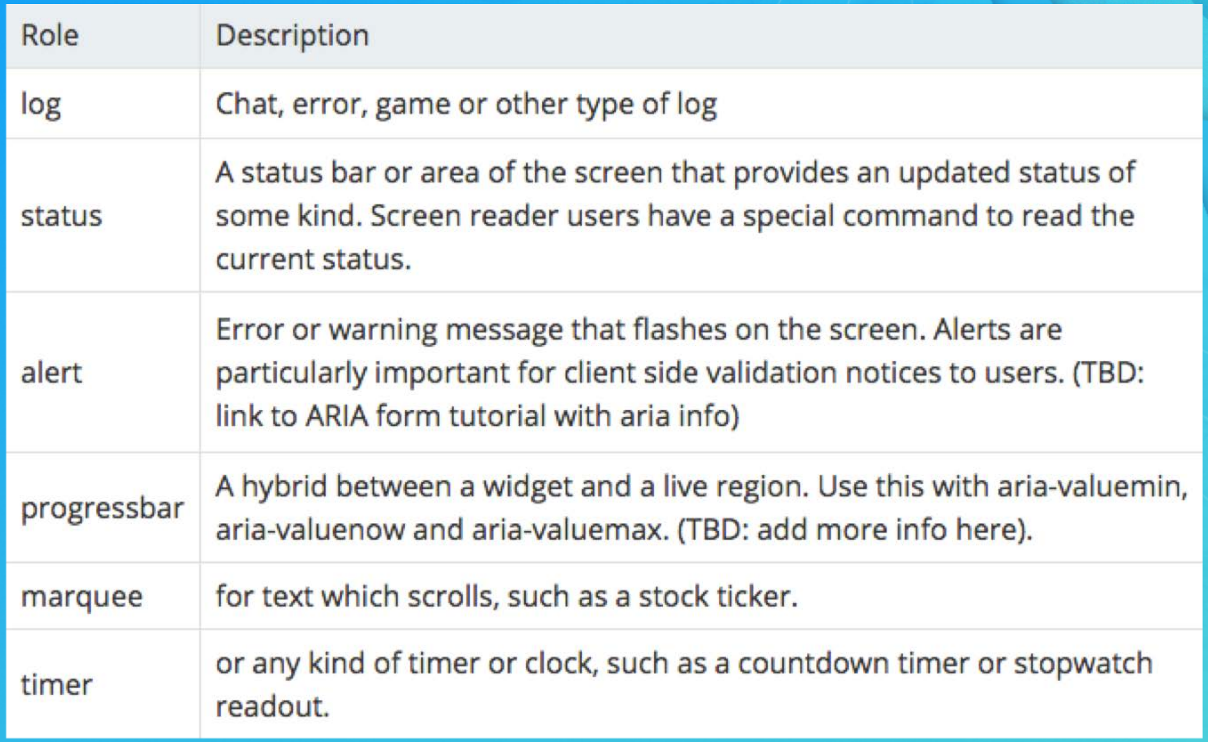
```
<label for="firstname">First name:</label>
<input type="text” name="firstname” id="firstname”>
```

Use `aria-label`

```
<input type="text” name="search” aria-label="Search">
<button type="submit">Search</button>
```

Live region

- Ask screen reader to announce dynamic changes
- Roles for various contexts



Role	Description
log	Chat, error, game or other type of log
status	A status bar or area of the screen that provides an updated status of some kind. Screen reader users have a special command to read the current status.
alert	Error or warning message that flashes on the screen. Alerts are particularly important for client side validation notices to users. (TBD: link to ARIA form tutorial with aria info)
progressbar	A hybrid between a widget and a live region. Use this with aria-valuemin, aria-valuenow and aria-valuemax. (TBD: add more info here).
marquee	for text which scrolls, such as a stock ticker.
timer	or any kind of timer or clock, such as a countdown timer or stopwatch readout.

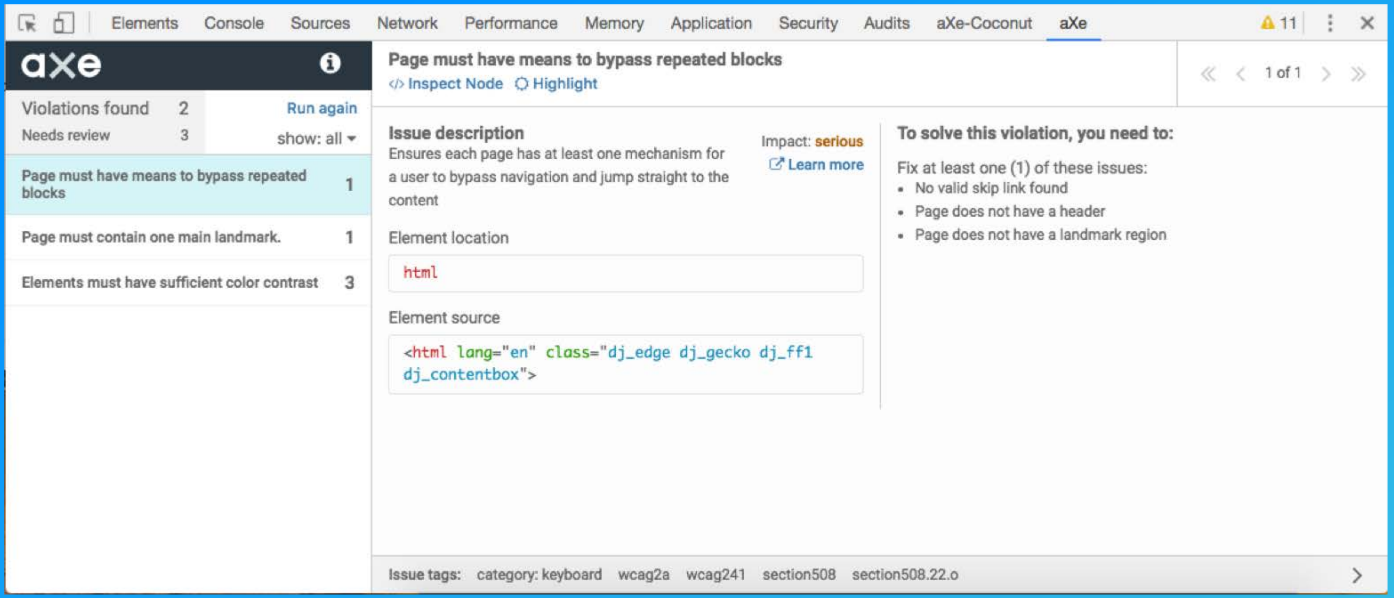
Test info and relationships

- Run automated test
- Use screen reader to tab through page
- Look for Skip-Navigation and landmarks
- Click on label to see if associated form input is focused
- Use screen reader to fill in forms

Testing

The process

- Automated test
- Keyboard test
- Screen reader test



Violations found 2

Needs review 3

Page must have means to bypass repeated blocks	1
Page must contain one main landmark.	1
Elements must have sufficient color contrast	3

Page must have means to bypass repeated blocks

Issue description

Ensures each page has at least one mechanism for a user to bypass navigation and jump straight to the content

Impact: serious

Element location

```
html
```

Element source

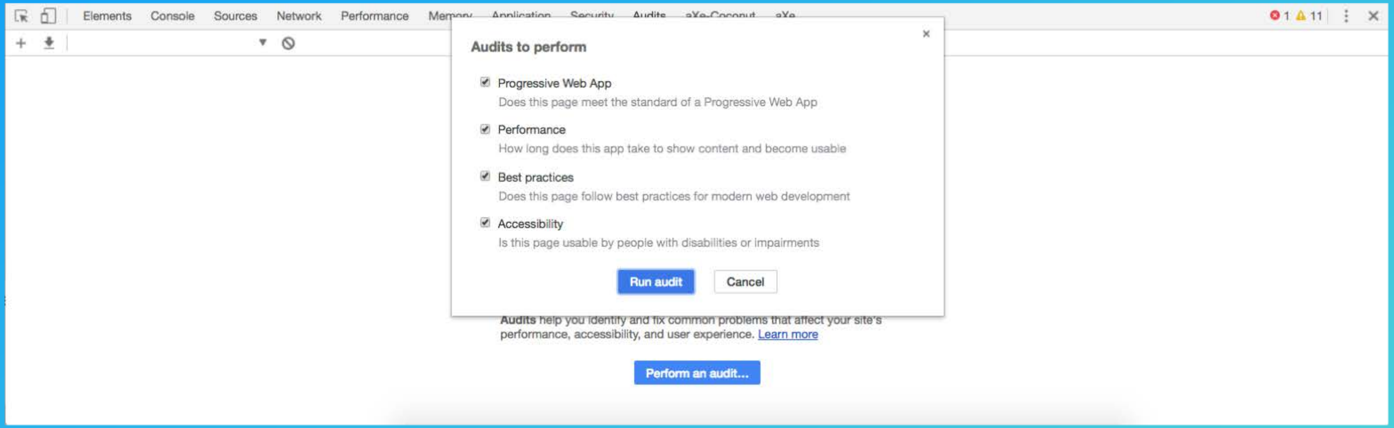
```
<html lang="en" class="dj_edge dj_gecko dj_ff1
dj_contentbox">
```

To solve this violation, you need to:

Fix at least one (1) of these issues:

- No valid skip link found
- Page does not have a header
- Page does not have a landmark region

Issue tags: category: keyboard wcag2a wcag241 section508 section508.22.o



Audits to perform

☑ Progressive Web App

Does this page meet the standard of a Progressive Web App

☑ Performance

How long does this app take to show content and become usable

☑ Best practices

Does this page follow best practices for modern web development

☑ Accessibility

Is this page usable by people with disabilities or impairments

Audits help you identify and fix common problems that affect your site's performance, accessibility, and user experience. Learn more

aXe

- Tests rendered browser DOM
- Less false positives
- Accessible
- Good documentation

Keyboard navigation

tab   shift tab

Move keyboard focus

enter

Click links

enter   space

Click buttons

↑

←   →

↓

Menus and some form controls

Screen reader

Recommended combinations:

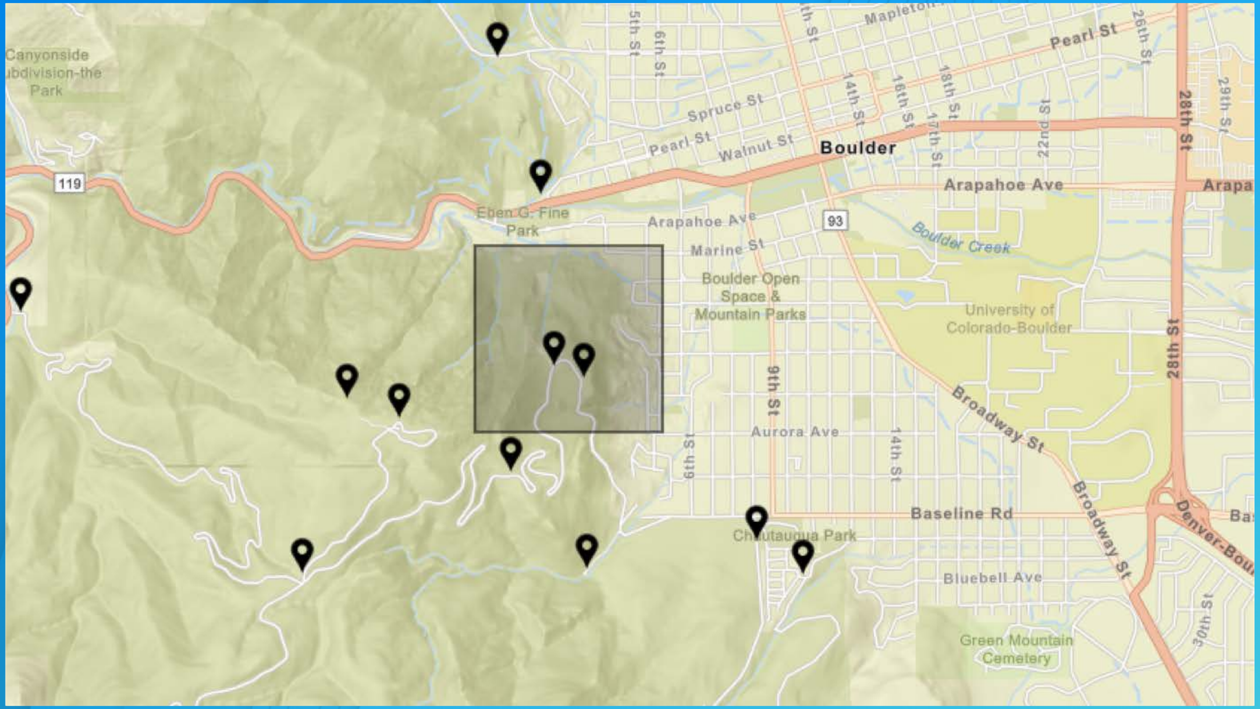
OS	Screen reader	Browser
MacOS	VoiceOver	Safari
Windows	NVDA	Firefox
Windows	JAWS	IE/Edge

Screen reader

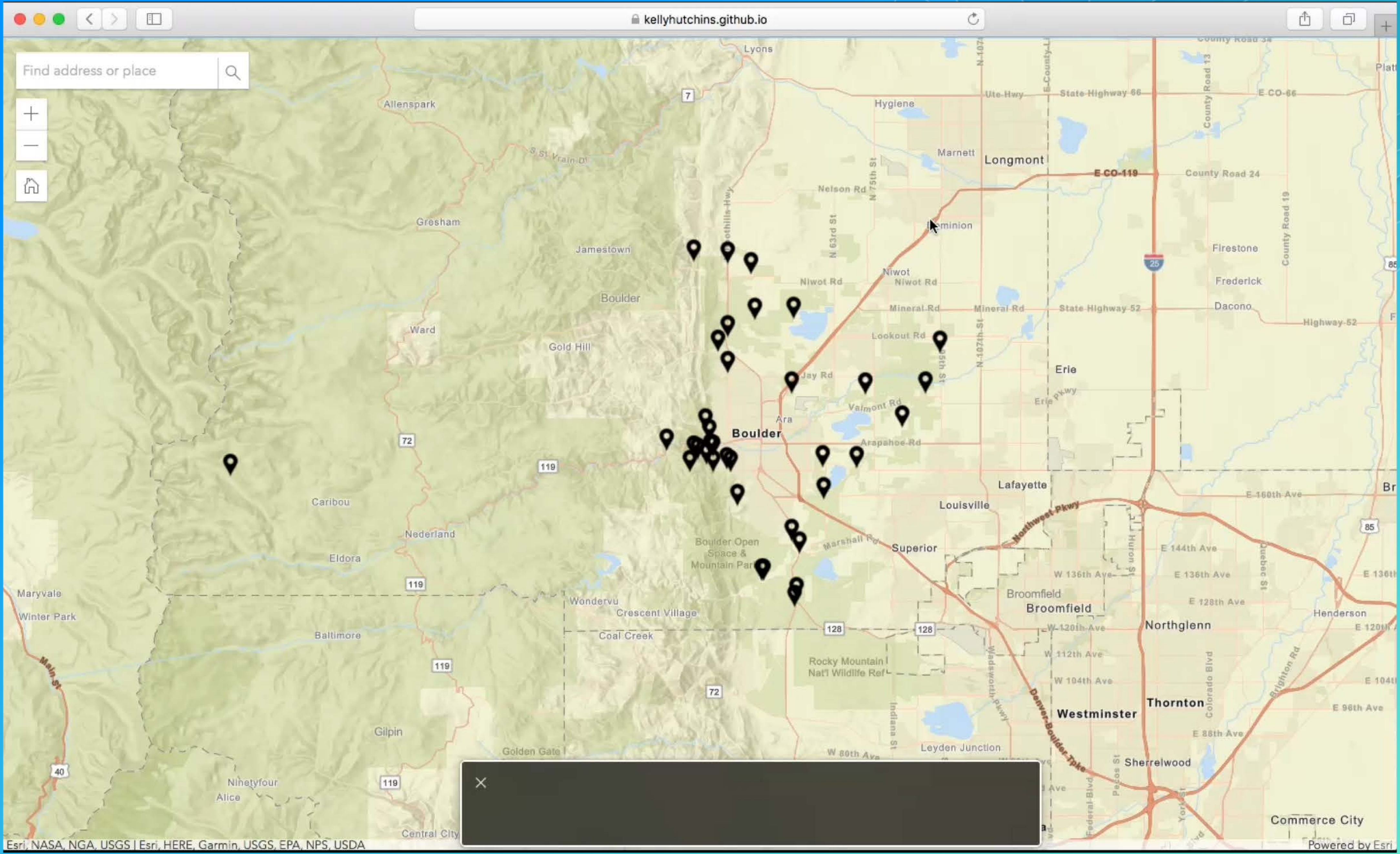
	Turn on	Turn off	Modifier key
VoiceOver	Command + F5	Command + F5	Control + Option
NVDA	Control + Alt + N	NVDA + Q	Numpad Insert
JAWS	Control + Alt + J	Insert + F4	Numpad Insert

VoiceOver commands

VO + right/left arrow	Read next/previous item
Control	Stop reading
VO + space	Click link, button, form controls
VO + u	Open rotor

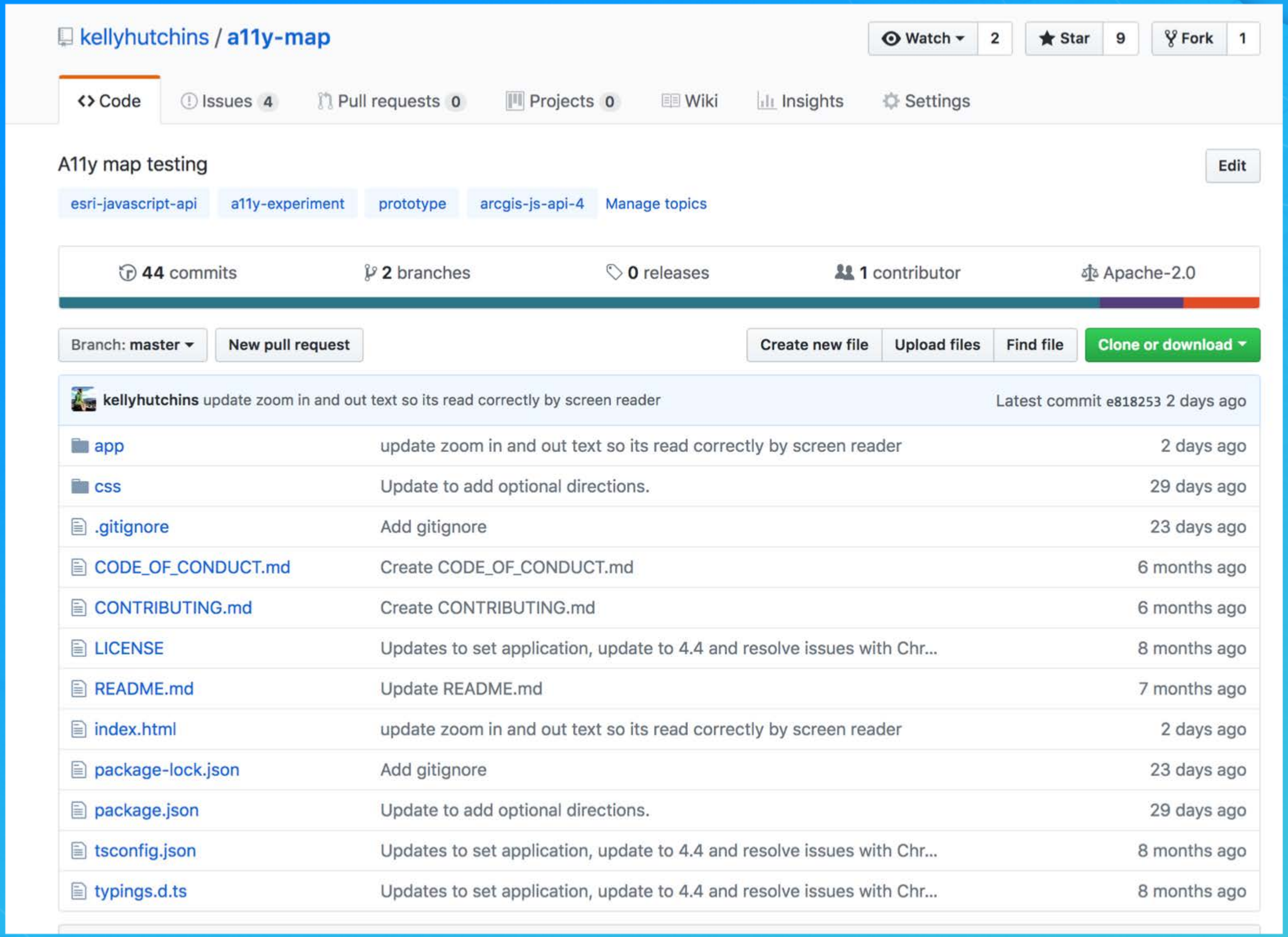


Accessible map



Esri, NASA, NGA, USGS | Esri, HERE, Garmin, USGS, EPA, NPS, USDA

Powered by Esri



A11y map testing

esri-javascript-api · a11y-experiment · prototype · arcgis-js-api-4

44 commits · 2 branches · 0 releases · 1 contributor · Apache-2.0

kellyhutchins update zoom in and out text so its read correctly by screen reader — Latest commit e818253 2 days ago

app	update zoom in and out text so its read correctly by screen reader	2 days ago
css	Update to add optional directions.	29 days ago
.gitignore	Add gitignore	23 days ago
CODE_OF_CONDUCT.md	Create CODE_OF_CONDUCT.md	6 months ago
CONTRIBUTING.md	Create CONTRIBUTING.md	6 months ago
LICENSE	Updates to set application, update to 4.4 and resolve issues with Chr…	8 months ago
README.md	Update README.md	7 months ago
index.html	update zoom in and out text so its read correctly by screen reader	2 days ago
package-lock.json	Add gitignore	23 days ago
package.json	Update to add optional directions.	29 days ago
tsconfig.json	Updates to set application, update to 4.4 and resolve issues with Chr…	8 months ago
typings.d.ts	Updates to set application, update to 4.4 and resolve issues with Chr…	8 months ago

https://github.com/kellyhutchins/a11y-map

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

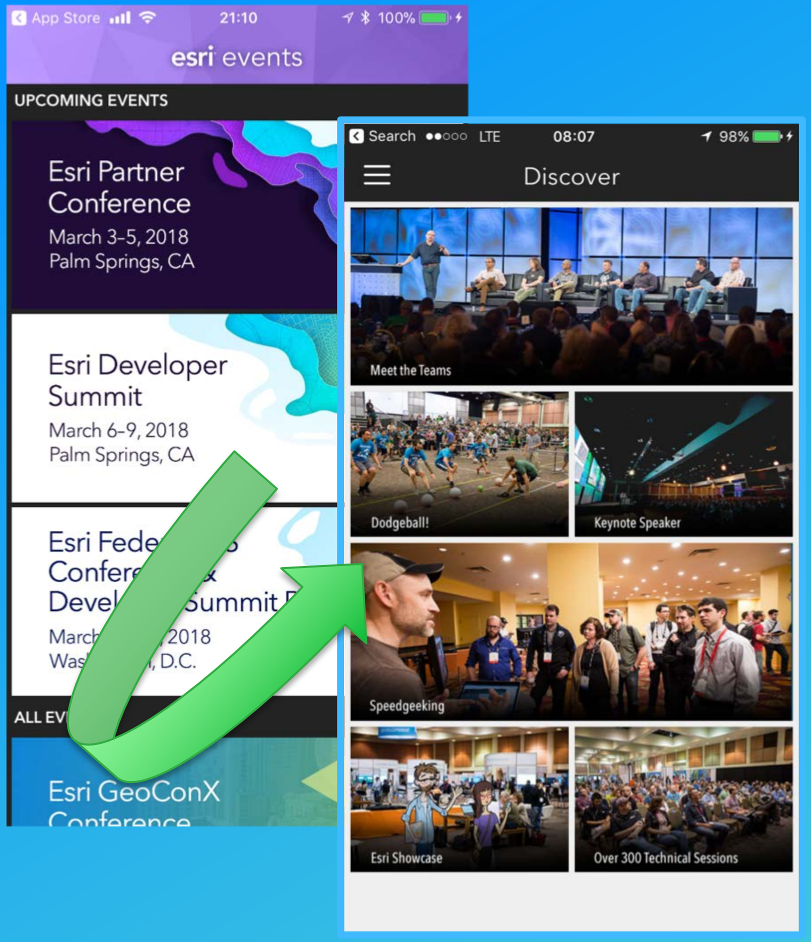
Friday, March 9

8:30 – 9:30 AM
Accessible Web Mapping Apps

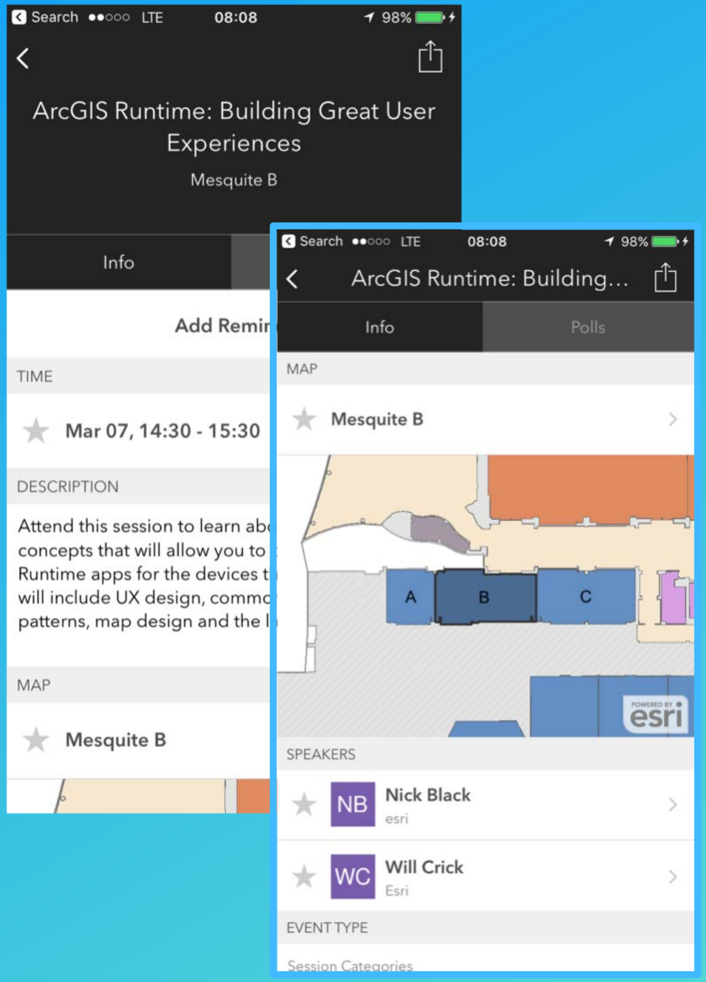
10:00 – 11:00 AM
DIY Usability Testing
Smoketree A-E

Please Take Our Survey!

Download the Esri Events app and find your event



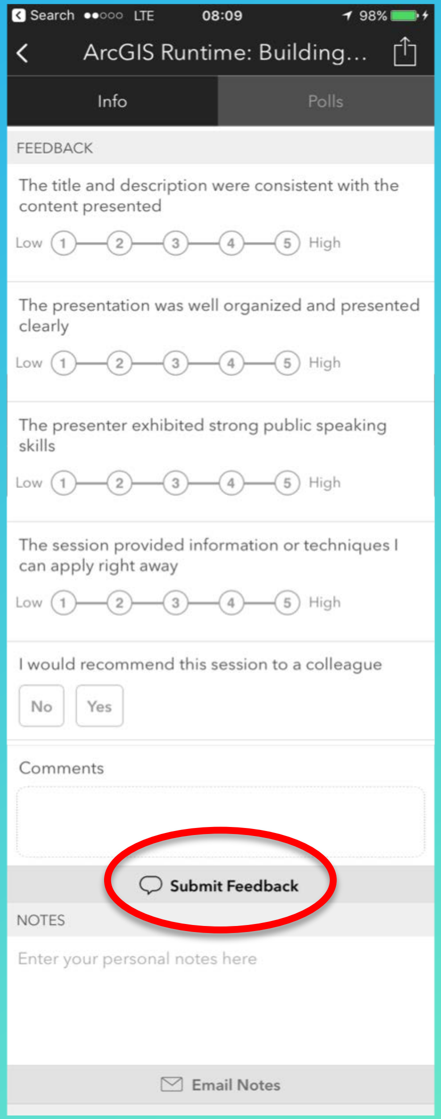
Select the session you attended



Attend this session to learn ab
concepts that will allow you to
Runtime apps for the devices t
will include UX design, commo
patterns, map design and the l

Scroll down to the “Feedback” section

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



The title and description were consistent with the content presented

The presentation was well organized and presented clearly

The presenter exhibited strong public speaking skills

The session provided information or techniques I can apply right away

I would recommend this session to a colleague esri

THE
SCIENCE
OF
WHERE