Presenter notes

What is accessibility

Accessibility is a quality. It means how easily and effectively a product or service can be accessed and used.

Good accessibility is designed for the full range of capabilities, as well as for the context of use or environmental constraints.

Wide range of disabilities

"Disabilities" could be permanent, temporary, or situational. For example, a parent may be holding a child while trying to use the phone; or a person may be in a noisy environment while trying to watch a video. People's normal abilities may be constrained by the situation, but they still need the same access to information.

Why is accessibility important

At some point in our lives, disability will affect most of us, not matter who we are, especially as we get older.

Everyone has abilities and limits to those abilities. Those limits should not be treated as obstacles to accessing the web.

Great design starts by thinking about how to make products work for everyone.

WCAG

The Web Content Accessibility Guidelines (WCAG) is developed by the W3C. It is the international standard for making web content accessible. WCAG has been adopted by laws and standards in a lot of countries and regions. In the US, the Section 508 regulations incorporates WCAG 2.0.

WCAG 2.0 was published in December 2008 and WCAG 2.1 was published in June 2018. WCAG 2.1 extends WCAG 2.0 by adding a number of new success criteria and guidelines.

Focus and keyboard accessibility

WCAG 2.4.7 Clear focus

Focus is the visual indicator that shows which control on the screen currently receives input from the keyboard. If users use keyboard to navigate a page, they should be able to see which control they are on at any time.

The best way to ensure clear focus is to have a visible border or outline around the element in use, whether it’s a form field, link, menu, content area or any other interactive element. This makes it easier for your users to understand where they are on a page.

Browsers have default styles for focus. We could customize the focus style if needed, but don't just disable the browser's default focus style.

WCAG 1.3.2 Tab order

Tab order is the order in which focus proceeds forward and backward through interactive elements via the Tab key. As a keyboard user navigates through the page, the order in which interactive items receive keyboard focus is important.

The default keyboard navigation order must be logical and intuitive. For example, in English we read from left to right and from top to bottom. So the tab order should follow the visual flow of the page: left to right and top to bottom. And we should let the web page's source code determine the tab order.

I should point out that letting the keyboard focus jumping around on the page is not only an accessibility issue, but also a usability issue. It just makes using the keyboard to work with the page a lot harder than necessary.

WCAG 2.1.1 Keyboard access

Many people use keyboard to use a web page. Some of them may have motor impairment, and some may be power users that are faster with keyboard. Like mouse users, keyboard users should be able to use forms, menus, or any other functionalities on a web page.

WCAG 2.1.2 Keyboard trap

A keyboard trap occurs when a user can get into a subsection on a web page by using the keyboard, but the user cannot get out of that subsection through the keyboard.

A common exception of this rule is when we show a modal dialog to the user, the dialog needs to trap keyboard focus until the user chooses to close the dialog. This is because the dialog introduces a new workflow to the user and needs the user's input to complete the new workflow.

Color and color contrast

WCAG 1.4.1 Color only info

Users with visual impairment need help when we use color on the web. As many as 1 in 12 men have some degree of color blindness. That means about 8% of male users may struggle if we don't use color correctly. For example, if we use red text only to show an error message, users who cannot see red will not notice that message is an error. So that error message needs at least one more means to convey the meaning, like an error icon.

As another example, when we use color only to show different datasets in a chart, color blind users may not understand the different datasets. We should add clear labelling and patterns to those datasets, in addition to colors.

This guideline is not about removing color from our web pages or making web pages black and white, far from it. It just means that color should not be the only way of conveying information. Adding additional ways of conveying the same information in a lot of cases not only improves the accessibility, but also the overall user experience.

WCAG 1.4.3 Contrast ratio

All of us will benefit from a good contrast between text and its background color. For users with visual impairments, good contrast is essential.

White text on colored background

When using white text on a colored background, it is sometimes hard to achieve a good contrast. You may be surprised how dark the color needs to be to meet the 4.5 contrast ratio. This can create hierarchy issues when those elements are not supposed to be the focus of the page - dark colored backgrounds will really grab the user's attention.

Flipping the contrast

You can solve the problem by flipping the contrast. Instead of using light text on a dark colored background, use dark colored text on a light colored background. The color is still there to help support the text, but it's way less prominent and does not interfere as much with other parts of the page.

Text alternatives

Alternative (or `alt`) text is a text replacement for non-text content like images. Non-sighted users rely on the alternative text to understand information in images. When they use screen reader and encounter an image, the screen reader will announce the alt text of the image. If the image does not have alt text, the screen reader will usually announce the file name of the image, which is not useful.

A common exception for this guideline is that if an image is decorative and it does not convey meaning, we should set the alternative text to an empty string.

Test for accessibility

We recommend a four-step process to test accessibility of web applications. The four steps are automated test, keyboard test, screen reader test, and color test.

This is also the process we follow at Esri. It breaks down the testing to a logic flow, and it minimizes the switching between different types of tests that we need to do.

Automated test with aXe

aXe is an extension that you can install in Chrome and Firefox. It tests the rendered page in Chrome and Firefox. It reports any violations against a set of accessibility rules and best practices. A big advantage of aXe, compared to other automated testing tools, is that it is relatively conservative, meaning it tends to report less false positives. And aXe itself is accessible. Keyboard users and non-sighted users should be able to use aXe for automated tests.

(After demo:) I should point out that automated tests cannot replace manual tests using keyboard and screen reader. Automated test tools like aXe can help us quickly find certain types of issues, but not all. For example, an automated test tool can tell us an image is missing alt text, but it cannot judge whether an alt text is meaningful for the image.

Keyboard test

The second step is keyboard test. This means using the Tab key or Shift and Tab keys to move focus on the page, using Enter key to click links, and using Enter or Space key to click buttons. Some interactive elements, like menus, tabs, or select, may require the arrow keys.

The goal of the keyboard test is to see if the web page provides the same functionalities to keyboard users as mouse users. It may help to unplug or disable the mouse, and try to use the keyboard to perform the tasks that mouse users would do. If an interactive element does not have keyboard focus or it cannot be tabbed to, there is a keyboard accessibility issue. If the focus jumps around on the page, there could be a keyboard accessibility issue as well.

Screen reader

The third step is testing using screen readers.

A "screen reader" is the generic term for a program that helps blind people use a computer. Simply put, a screen reader will "read" (speak) the content of a page to the blind user.

A blind person using a screen reader experiences a website linearly, a little bit at a time. If an element does not have enough information for the screen reader to read what it is, the blind user may have difficulty understanding what the element is meant for.

Due to time limitation, we will not discuss details of using screen readers. But here is recommended pairing between screen readers and browsers. VoiceOver is the built-in screen reader in MacOS, so it works well with Safari. NVDA is a free screen reader in Windows and it is recommended for Firefox. Jaws is another screen reader in Windows and traditionally it works well with Microsoft browsers like IE and Edge. Jaws requires a paid license.

Screen reader basics

We recommend these two video tutorials for VoiceOver and NVDA. They cover the basic commands to read and interact with web content for testing purposes. Both of the videos are parts of the a11ycasts series on Youtube. We will share this presentation so you will have the URLs.

Color test

The last step is manual color test. This step is necessary because automated test tools cannot cover all color issues. For example, a background may use color gradient, not a single color. And for information like error messages, we have to manually look at it to see if they use color only to communicate the meaning.