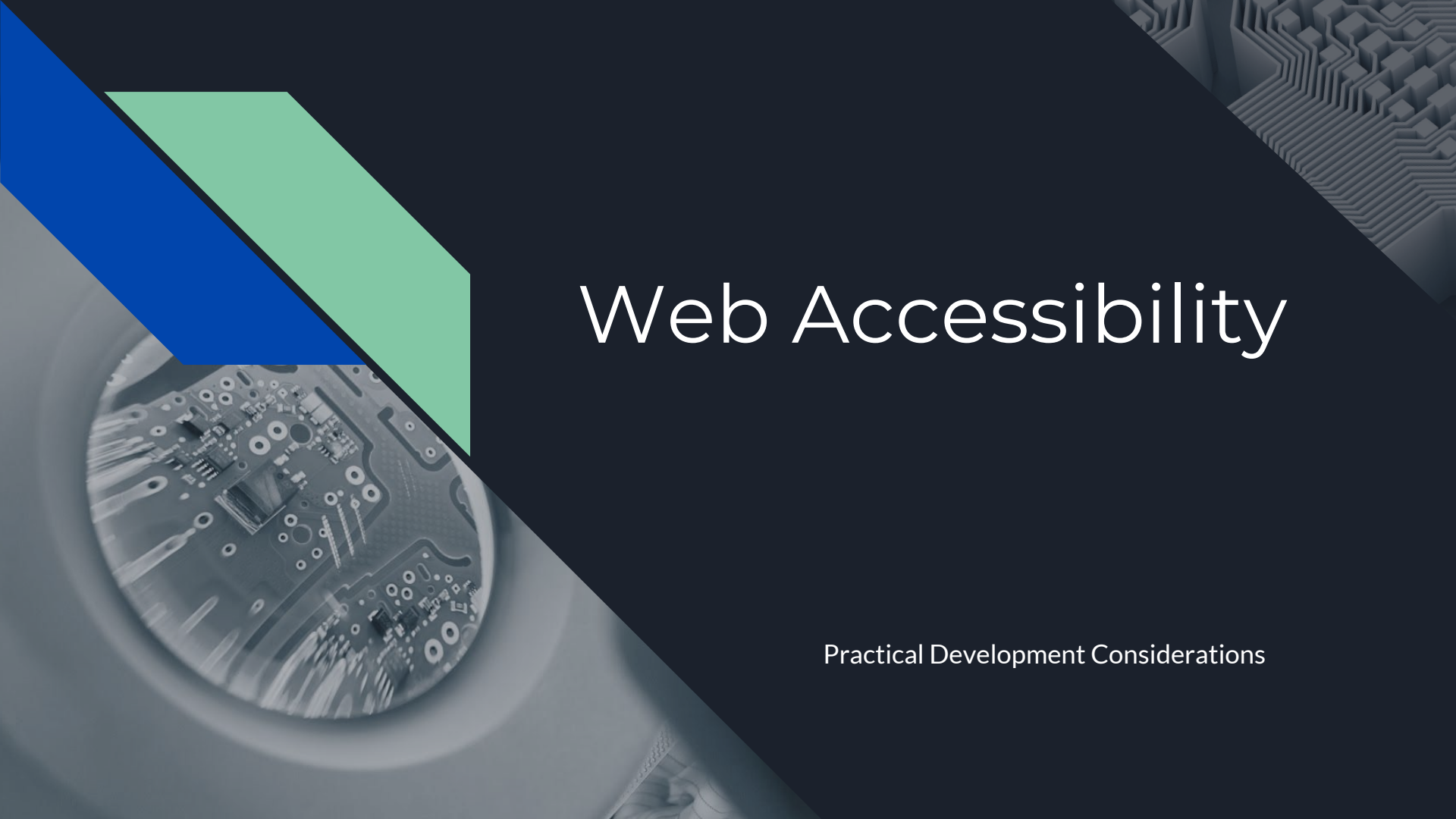


Web Accessibility

Practical Development Considerations

Table Of Contents

What Is Accessibility?

"Over a billion people, about 15% of the world's population, have some form of disability" - World Health Organisation.

Accessibility is about treating everyone the same, and giving them the same opportunities, no matter what their ability or circumstances.

For example, it is not right to exclude someone from a physical building because they are in a wheelchair, so parking spaces, ramps and lifts are generally provided to help.

We are all different, but we are all human, and therefore have the same human rights.

What Is Web Accessibility?

Web Accessibility is about giving a fair usage experience to differently abled users utilising a range of input devices.

It means that websites, tools and technologies are created in such a way so that all people can use them reasonably.

It allows all people to perceive, understand, navigate, interact and contribute to the web.

Catering For Different Disabilities

Many people are familiar with the JAWS or NVDA screen readers, but Web Accessibility caters for many types of disabilities including:

- Auditory
- Cognitive
- Neurological
- Physical
- Speech
- Visual

For example, Web Accessibility covers blindness, colour blindness, reading age and navigation of content for all types of user. It also includes improving experiences for different inputs including mobile and smart devices.

Web Accessibility Standards- Overview

WCAG (Web Content Accessibility Guidelines) are the Web Accessibility Standards that are used to “grade” how accessible a website is.

The guidelines are split into 3 basic levels: **A**, **AA**, **AAA**, with **A** being lowest and **AAA** being the highest certification of how accessible a website is.

As might be expected **AAA** standard is the most difficult to achieve, but not impossible. As a rule of thumb **AA** standard is accepted as a very good commitment to accessibility and is the general standard we should strive to aim for. We should also aim for all the **AAA** criteria that can be reasonably met, but what matters most is the website is gives the fairest and best experience to all users.

The levels break down to include some of the details listed on the following pages.

Web Accessibility Standards - A Grade

Examples of **A** grade criteria include:

-**Non-Text Content**: Must have a text alternative.

-**Use Of Colour**: Must not be the only means of conveying information.

-**Keyboard**: Must be able to operate all functionality of the content (with exceptions).

-**Focus Order**: Must preserve meaning and operability.

Web Accessibility Standards - AA Grade

Examples of **AA** grade criteria include:

-**All A Grade criteria**

-**Contrast Ratio**: Must be at least 4:5:1 (with exceptions).

-**Focus Visibility**: Keyboard focus indicator must be obviously visible.

-**Images Of Text**: Descriptions of the images must be present.

-**Text Size**: Must be able to resize up to 200% without assistive technology and without loss of functionality.

Web Accessibility Standards - AAA Grade

Examples of **AAA** grade criteria include:

-All A and AA Grade criteria.

-**Enhanced Contrast**: Must be at least 7:1 (with exceptions).

-**Keyboard (No Exceptions)**: All functionality of the content is operable with a keyboard without requiring specific timings for individual keystrokes.

-**Reading Level**: Content must be explained to a lower secondary level.

-**Change On Request**: Accessibility changes are initiated only by user input.

No ARIA at all is better than badly implemented ARIA

Semantic Markup

Semantic markup is critical before utilising ARIA properties. No ARIA at all is better than badly implemented ARIA.

Incorrect ARIA misrepresents visual experiences to screen reader users, potentially changing the meaning of the content and confusing the user.

-An ARIA role is a promise that code has been written to handle accessible interactions.

-ARIA properties can both hide and enhance functionality- adding (additional) meaning to content to support assistive technologies such as screen readers.

KEY POINT: ARIA roles and properties, generally, do not provide code for assistive technologies and require Javascript to be supplemented to enhance the experience correctly.

Link And Button Differences

Context. Links and buttons perform separate actions and are semantically different.

It is important that both the appearance and role of an element match the function it provides.

Links take the user to a destination. Buttons perform an action.

Elements can occasionally have the visual style of a link but perform the action of a button. In these cases we can give the link the role of button to help provide context for accessibility. A better solution, however, would be to change the visual style of the element so it matches the function and ARIA role.

Image Accessibility

All img elements should use an *alt* attribute to describe the image, this is a requirement of the HTML standard.

Context is critical in describing images and may differ greatly depending on the content and surroundings of the image itself.

A short and precise description is generally all that is required.

KEY POINT: Do not use “image of…” in the alt text. It will usually be obvious to the user that the element is an image.

Forms And Inputs

Inputs should have a label element associated to it whenever possible to accurately describe the control. This can be done by adding a *for* attribute to the label that exactly matches the *id* attribute of the input.

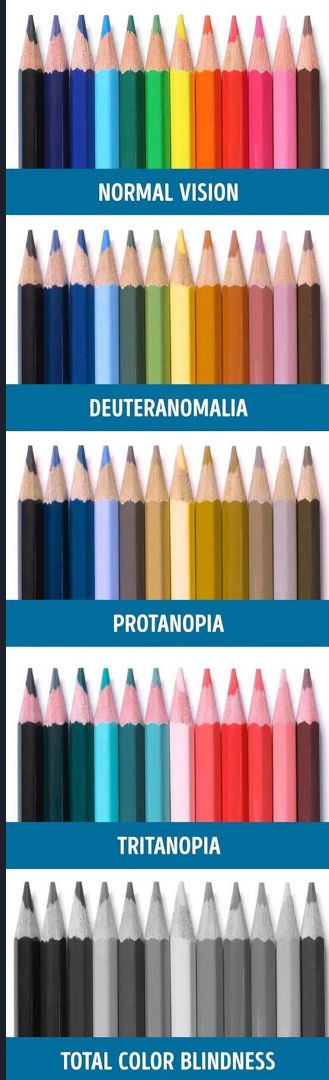
When a label element cannot be used, possibly because of a design decision, then label text can still be provided and associated to the input via the *aria-label* or *aria-labelledby* properties.

The overall aim should be to maintain a close and distinct relationship between the label and the input when describing the control.

Colour Accessibility

For a web page to have colours that are accessible for all users it must have a minimum contrast ratio of between 4:5:1 for **AA** standard and 7:1 for **AAA** standard.

Colours should not be the sole indicators of key information, for example *errors* and *warnings*, so should be coupled with text to help describe the visuals.



Showing And Hiding Content

Content should not be shown or hidden unexpectedly, for example when interacting with a checkbox.

Content that is hidden with *display: none;* is essentially invisible to all users. When that content is revealed with *display: block;* it visually appears for sighted users but screen reader users have no indication that the page content has visually changed.

If showing and hiding content, it should be made fair for all users with ARIA properties such as a *aria-describedby* to provide a description of the action and *aria-live* to announce a change in the page structure.

Disabled Attributes

Depending on context, *disabled* attributes, with no aria equivalent, should not be used as it effectively renders the disabled input invisible for screen reader users.

The *disabled* and *aria-disabled=”true”* properties should be coupled so that an input can be disabled, as normal, and also detected by screen readers.

Focus

Focus helps to give users a visual indicator of where they are on a page.

Focus should be obviously and visually represented to assist users who may only be navigating via a keyboard or input other than a mouse.

Focus should never be changed unnecessarily or unexpectedly, and any actions that adjust focus should be well described.

Summary

Web Accessibility is not easy nor is it a throwaway term- it is fairly subjective and the different accessibility agencies can pass or fail certification on seemingly random criteria.

ARIA properties should not be abused or overused, they should enhance the experience for all users to give a fair experience. No ARIA at all is better than badly implemented ARIA.

AA certification is very achievable and a good level of accessibility can be attained with only standard semantic markup if it is written with accessibility in mind.

In conclusion, web content should always be created with accessibility considered to give a good experience to all, even if it isn't explicitly required or asked for.

Resources And Further Reading

WCAG Outlines: https://www.w3.org/WAI/WCAG21/quickref/

WAI ARIA Examples: https://www.w3.org/TR/wai-aria-practices/examples/

Digital Accessibility Centre: http://digitalaccessibilitycentre.org/

WUHCAG: https://www.wuhcag.com/wcag-level-aaa/

Accessible Web: https://accessibleweb.com/

WebAIM: https://webaim.org/

Deque: https://www.deque.com/

MDN: https://developer.mozilla.org/en-US/docs/Learn/Accessibility

Any Questions?