

# Accessibility: At the forefront of our minds



Jack Domleo  
Front-end Developer

@jackdomleo7



# What will you get out of this workshop?

- The **true** meaning of “What is accessibility?”
- The consequences of inaccessible websites & applications
- Get into the habit of writing in an accessible manner
- Think accessibility first
- How to test accessibility
- Some really good resources to start learning accessibility

What is accessibility?

Accessibility is also known as A11y  
(pronounced “Ally”)

The diagram illustrates the shorthand 'a11y' for 'accessibility'. The word 'accessibility' is shown in a serif font. The first letter 'a' is black, and the following 11 letters 'ccessibilit' are red. Below the red letters, numbers 1 through 11 are placed, each centered under a letter. A horizontal bracket spans from the first red letter 'c' to the last red letter 't'. An arrow points down from the center of this bracket to the word 'a11y' below. In 'a11y', the 'a' is black and the '11' is red, matching the color scheme of the original word.

a c c e s s i b i l i t y

1 2 3 4 5 6 7 8 9 10 11

a11y

# Who is accessibility for?

**Everyone!**

“Accessibility is for everyone so blind users can use the website.” - *NEARLY*

“Accessibility is for everyone, regardless if they have a disability or not, regardless of the device they are using or the preferences they have set.” - *MY DEFINITION*

<https://www.w3.org/WAI/fundamentals/accessibility-intro>

# Accessibility Myths

1. Accessible websites are ugly and boring
2. Making a website accessible is expensive, time-consuming and hard - it's not profitable
3. Accessibility only caters for a small percentage of people
4. Web accessibility is optional
5. Automated accessibility evaluation tools are enough
6. Websites are either accessible or inaccessible
7. Thinking accessibility is down to the developers

# Levels of Accessibility

- WCAG A
- WCAG AA (Aim for this as a bare minimum at all times!)
- WCAG AAA

# What makes a web page accessible?

- Responsive Design
- WCAG AA / AAA Compliant Colours
- Clear User Interface (UI)
- Simple User Experience (UX)
- Meaningful markup
- Correct spelling
- Clear grammar & content
- Keyboard Interactive
- Consistent Terminology

...to name a few



# Nothing is 100% accessible | WCAG AAA is hard

We cannot cover every single edge case of every user that will ever exist that will visit our web page.

Aiming for WCAG AAA is great and achievable, however it requires a lot of time investment - WCAG AA should be the minimum aim and is often a lot less work, time and money invested to achieve.

*Depending on your industry/country/etc, there are certain laws around accessibility requirements - for example, [in the UK, all public sector websites and web applications must meet WCAG 2.1 AA](#)*

# Accessibility is **not** a true or false

Accessibility on a web page can't be measured by either:

- “Yes, my web page is accessible” - *TRUE*
- “No, my web page is NOT accessible” - *FALSE*

Instead, accessibility should be seen as a scale where we strive to trend closer and closer to being 100% accessible.

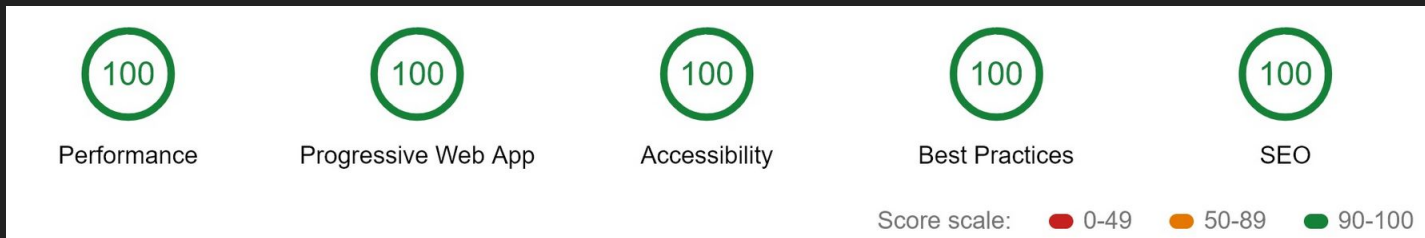
*E.g.* If you add missing alt text to an image, you have improved the accessibility of that web page, but it cannot be classed as being 100% accessible.

# The Chrome Lighthouse confusion

Chrome's Lighthouse feature is very clever, but it's not that clever for accessibility.

It's clever enough to spot if your input form fields are missing a label, or if your image is missing alt text, but it doesn't think like developers do.

*E.g.* If we create a custom dropdown component, Chrome doesn't know that this was your intention and therefore won't notify you if you've missed or misused any HTML semantic elements, role attributes or any of the aria attributes.



# I need accessibility! You need accessibility!

Accessibility needs are different for every individual.

Here are some of my needs & preferences:

- Stammer (speech recognition is progressing for speech impairments)
- Poor eyesight (colour contrast is important)
- Form tabbing (tabbing is much quicker for me to do)
- URL hint (requires correct use of `<a>` and `<button>`)



# Testing Accessibility

- Zoom in/out
- Disable CSS & JS - the web page should still be readable & understandable
- Resize the window / Use developer tools to test different devices
- Automated scanner such as Chrome's Lighthouse feature, axe, WAVE, Pa11y & Tota11y
- [Firefox's accessibility feature](#)
- Learn how to use a screen reader
- Real user feedback
- [Web Accessibility VS Code plugin](#)
- Unplug your mouse
- Cover your screen
- Turn on high contrast mode
- Turn off images & media
- Click/tap form labels
- Temporarily apply [Checka11y.css](#)

Automation testing is not be sufficient enough to test accessibility... yet - try creating a solid manual test plan for accessibility. Alternatively, this works well with exploratory testing.

# Consequences of Ignoring Accessibility

- Excluding a potentially large number of users from your web page:
  - Could lose a potential customer/sale
  - Customer may ask for refund
  - More potential calls/queries to the customer support team
  - Potential bad reviews
  - Lawsuit ([read about Domino's Pizza accessibility lawsuit](#))
  - Estimated between US\$35k and US\$38k to rectify to meet WCAG 2.1 AA
  - More work to do later

Justifying implementing accessibility is hard! If you have to justify implementing accessibility, it's already become an afterthought.



# Where to start learning accessibility?

Some great resources for anyone learning a11y at any level:

- The A11y Project - [www.a11yproject.com](http://www.a11yproject.com)
- Inclusive Components - [inclusive-components.design](http://inclusive-components.design)
- A11y-components - [github.com/t12t/a11y-components](https://github.com/t12t/a11y-components)
- Google search, “Common a11y mistakes”
- W3C - [www.w3.org](http://www.w3.org)
- “Accessibility at the Forefront of our Minds” talk by Jack Domleo 🗣️

# Is there such a thing as 'too accessible'?

Something to think about 🤔

Question 5: What is the below image an image of?

