Web Accessibility

The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.

Tim Berners-Lee https://www.w3.org/standards/webdesign/accessibility

Preface whole thing with I am not an expert!!
In principle who would disagree but a11y is often overlooked - its not baked into our AC/requirements so it becomes an afterthought. We do it but no consistently and often by accident.

Now building public facing stuff, although is important for non-public stuff as well.

No legal obligaition in the UK...yet...

Aims

- The principles underlying web accessibility
- How can we make accessibility issues more visible to us?
- What are the key things to consider?
- What steps can we take today to start moving in the right direction?

- it's invisible to us (obvious if css is broken, not obvious if a11y issues)
- there's a lot to consider and learn - overwhelming - different groups, different needs, standards are lengthy and dull etc

61% of devs who answered survey by edge team dev do no accestblity testing

Web accessibility encompasses all disabilities that affect access to the Web, including:

- auditory
- cognitive
- neurological
- physical
- speech
- visual

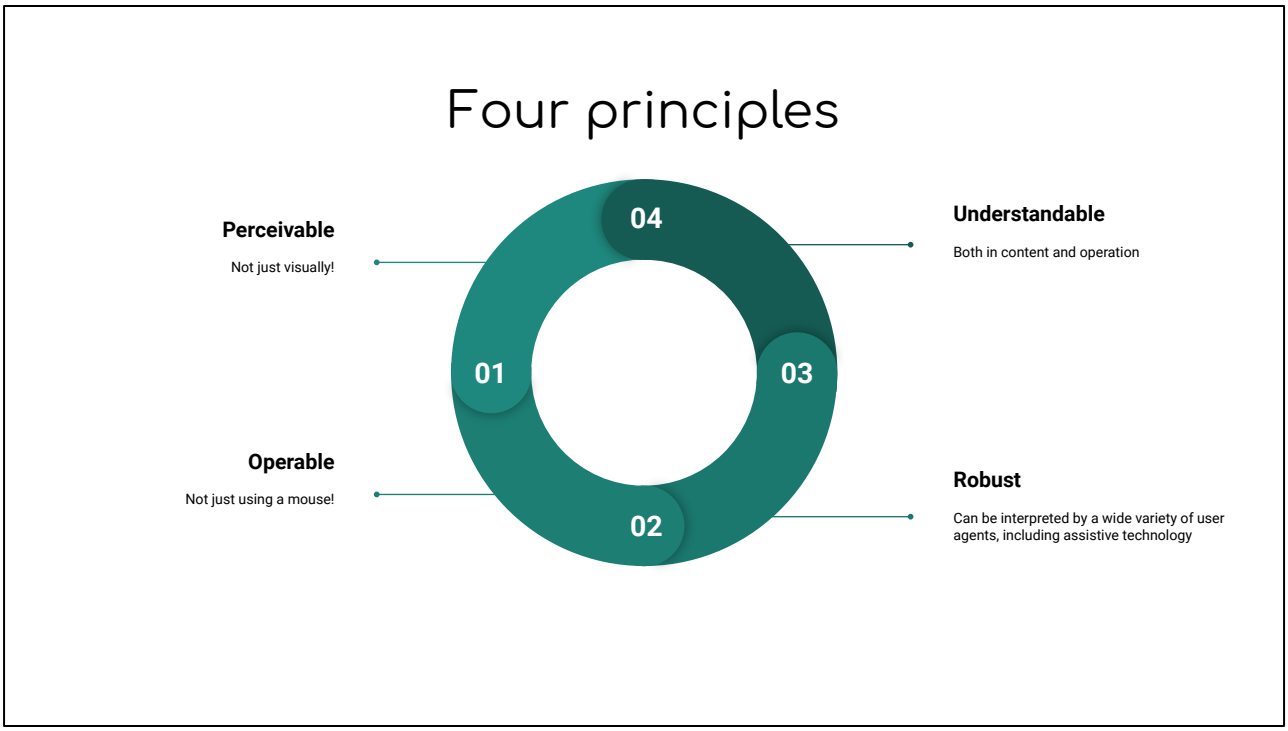
Web accessibility also benefits people *without* disabilities, for example:

- people using mobile phones, smart watches, smart TVs, and other devices with small screens, different input modes, etc.
- older people with changing abilities due to ageing
- people with "temporary disabilities" such as a broken arm or lost glasses
- people with "situational limitations" such as in bright sunlight or in an environment where they cannot listen to audio
- people using a slow Internet connection, or who have limited or expensive bandwidth

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

Something like 1 in 5 people in Uk has some type of disability - there's a wide range and wide range of resultant needs...which makes it even more daunting

But it benefits us all...and overlaps with seo so there's extra weight there to the business argument



wcag - web content accessibility guidelines

for each principle there are guidelines with success criteria, along with sufficient and advisory techniques

Tools

- eslint-plugin-jsx-a11y
- Axe
- Lighthouse
- NVDA (or Voiceover for Mac)
- Dev tools accessibility inspector

tools not a panacea - don't pick up everything - eg can automate some stuff but no current way of automating screen reader tests

Will use Axe a couple of times in this talk

DEMO DEV TOOLS

So, what are the key areas to think about?

As I said, there's a huge amount of stuff, far more than I can cover in an hour long talk...so will cover the key areas alhtough its not exhaustive (e.g. not going to talk about concerns around video and audio content and not going to cover some stuff you probably already know like alt tags and link text)

Contrast

- Use tools to review colour contrast (e.g Axe)
- This is one of several areas where design and dev need to work together

- Contrast: OK this is more design, and for some customers will be out of our controle to some extent - obviously a lot of accessibility is rooted in the design of the page so we need to work hand in hand, even though I'm obviously gonna focus more on the dev side

Gov.uk is a beacon of accessibility - simple design - but run axe...get colour contrast warnings

Oh, and don’t rely on colour alone to indicate things...colour-blindeness

Zoom

- Use the browser zoom
- Testers could test the zoom functionality on other devices

- Zoom: responsive design can help with this, try zoom see what happens (200%)...meta tags…

Guardian as example get over 200 and goes to mobile...

Focus Management

- Use focusable elements for interactive items (or use tabIndex = 0)
- Make sure focused element is signalled visually
- Ensure that focus is handled correctly for modals etc:
  - Their first interactive element should get focus
  - Navigation by keyboard should not result in ending up with focus outside the modal while it is open (focus trap)
- Hidden focusable elements should not be able to be tabbed to: toggle visibility:hidden
- Client side routing means need to consider where focus should go (and ensure the change gets announced)

Visibility cf keyboard nav
Also important for screen readers (announcing)

Gov.uk has skip links a fugly but visible style

Santander starts off well but rapidly get nothing visible

```
document.body.addEventListener('focusin', (event) => {
    console.log(document.activeElement)
})
```

Client side routing - no page refresh - less simple to solve, various solutions - Gatsby research - focus on a heading gave best experience for screen readers and having visible focus was crucial for speech recognition software
They also tests with users using zoom. Slightly different results for different techs illustartes the issue...
Conclusion was that using skip links (small non visible areas that get focus -> visible, can announce change and can allow navigation) was maybe best...perhaps with an aria live region that updates on client side navigation

Keyboard Navigation

- Can you navigate just using the keyboard?
- Keyboard interaction patterns are set out in the WAI-ARIA Authoring Practices so you don't need to guess! https://www.w3.org/TR/wai-aria-practices-1.1/

Semantic html and ARIA

- If at all possible use the appropriate html element
  - But some feature may not be implemented in html or you may have design considerations that cannot be accommodated using the native element…
- Aria allows information to be added to elements to allow assistive technology to interpret non-standard widgets etc
- First rule of aria-club: no aria is better than bad aria
  - Roles are a promise of certain behaviour
  - Aria will sometimes override the built-in accessibility semantics
- Aria comprises:
  - Roles - what it is
  - Properties - other immutable information e.g. if its is required
  - States - the current condition of the element

Accessible Rich Internet Applications

No aria better than bad aria - eg the broken promise of role="button"... webAIM million: sites with aria more likely to have accessibility issues!!

LOOK AT IMPLICAITONS OF ROLE=BUTTON

Structure and landmarks

- All pages have a structure and this should be reflected in the markup
  - For example using <main> for the main content
- Headings also form part of this
  - Should be using them in order
  - Shouldn't be using them just for visual impact

- in MySight we are more using h tags for visual differences - we aren't using them all and sometimes we skip them
  - information architecture - include how we approach alt text etc

Motion

- Autoplay remains evil
- prefers-reduced-motion
- Do you really need that animation?

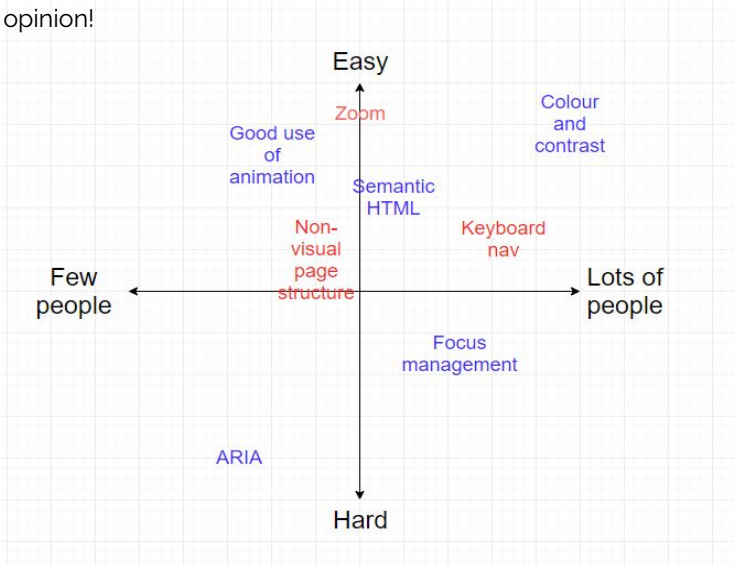
vestibular disorders/migraine/epilepsy/sensory processing disorders/adhd etc...also low quality/mobile devices

https://leedsdigitalfestival.org/

candidate recommedation for Update media query which would allow us to check device ability to update after render

Where should we focus our efforts?

A non-scientific opinion!



Screen readers as a litmus test - small % of users but if can be read by screen reader means its pretty accesible
Need to consider though not only aria but combos of software and browser so it not totally simple

What can we start doing now?

- Use the right html element for the job
- Navigate using your keyboard
- Zoom in
- Start using a tool to flag issues
- Building a component? Look at the authoring practices and/or projects like Downshift and Reach UI: bake accessibility into our base components
- Add accessibility issues to the mental list of things you look for on code reviews

Resources wiki page

Some stuff is more structural - AC/Requirements/Checklists/more integrated tooling, unit or integration tests/what user groups do we prioritise?
Some stuff is just harder...e.g. time to learn screen reader and researching how to best implement this stuff...Ideally - user testing/expert audit but manual testing is important (hard though if not an expert user of AT -see that gatsby results for how different solutions can impact users of AT differently)