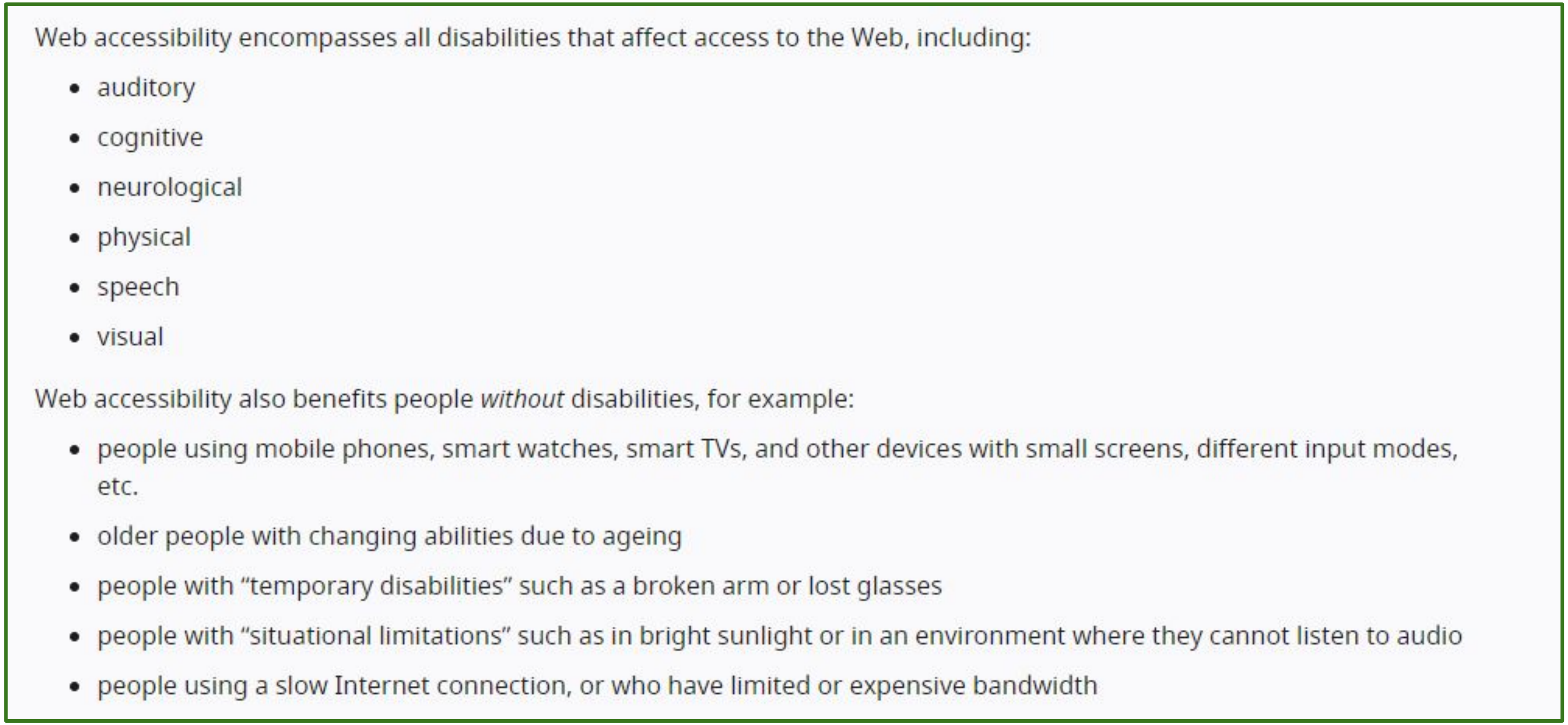
Web Accessibility

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


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:

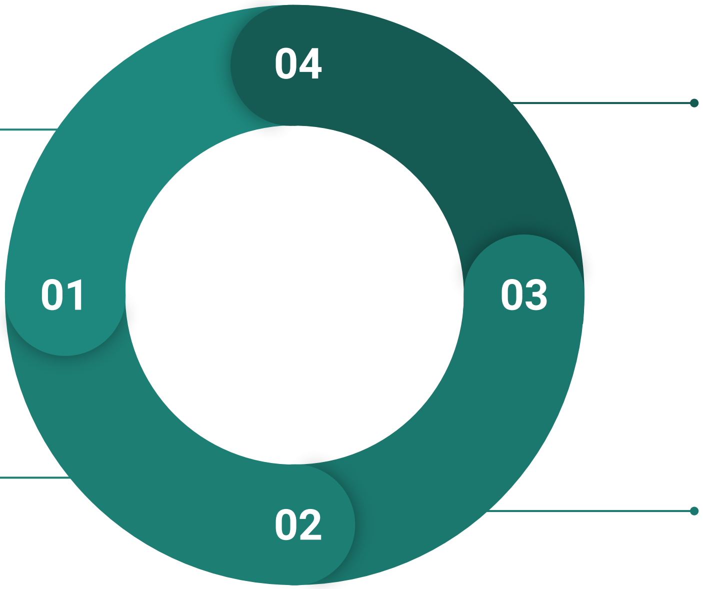
- people using mobile phones, smart watches, smart TVs, and other devices with small screens, different input modes, etc.
- older people with changing abilities due to ageing
- people with "temporary disabilities" such as a broken arm or lost glasses
- people with "situational limitations" such as in 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/

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?

Four principles



Perceivable

Not just visually!

Operable

Not just using a mouse!

Understandable

Both in content and operation

Robust

Can be interpreted by a wide variety of user agents, including assistive technology

Tools

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

So, what are the key areas to think about?

Contrast

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

Zoom

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

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

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

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

Motion

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

An example: Building a select element

Summary

- Contrast
- Zoom
- Focus Management
- Keyboard Navigation
- Semantic html and ARIA
- Structure and landmarks
- Motion

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