Accessibility

and why you should give a

DOM 🖥️
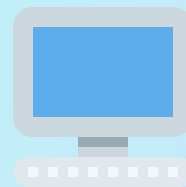
Our journey today

Common Questions

- ♿ What does it mean to be *accessible*?
- 🔍 What is screen reader?
- 🖥️ What is my role as a developer/designer/XM?
- 📚 What resources are there to help me build accessibly?

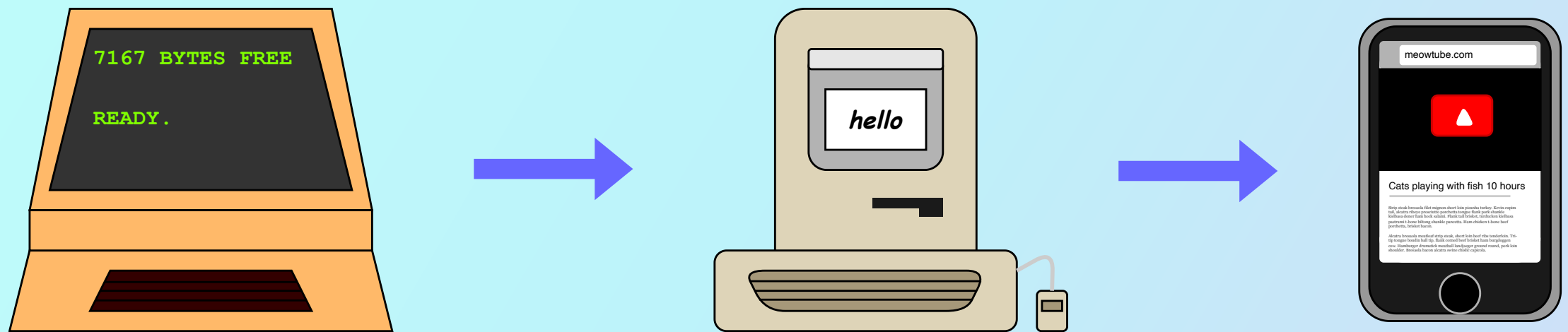# Accessibility

11 Characters

A-11-y

A11y

What makes something *accessible*?

What makes something *accessible*?

1977

The year of personal computing

TRS-80 **$600**

PET **$800**

Apple II **$1300**

Text-based interface

COMMODORE BASIC V7.0 122365 BYTES FREE
(C)1986 COMMODORE ELECTRONICS, LTD.
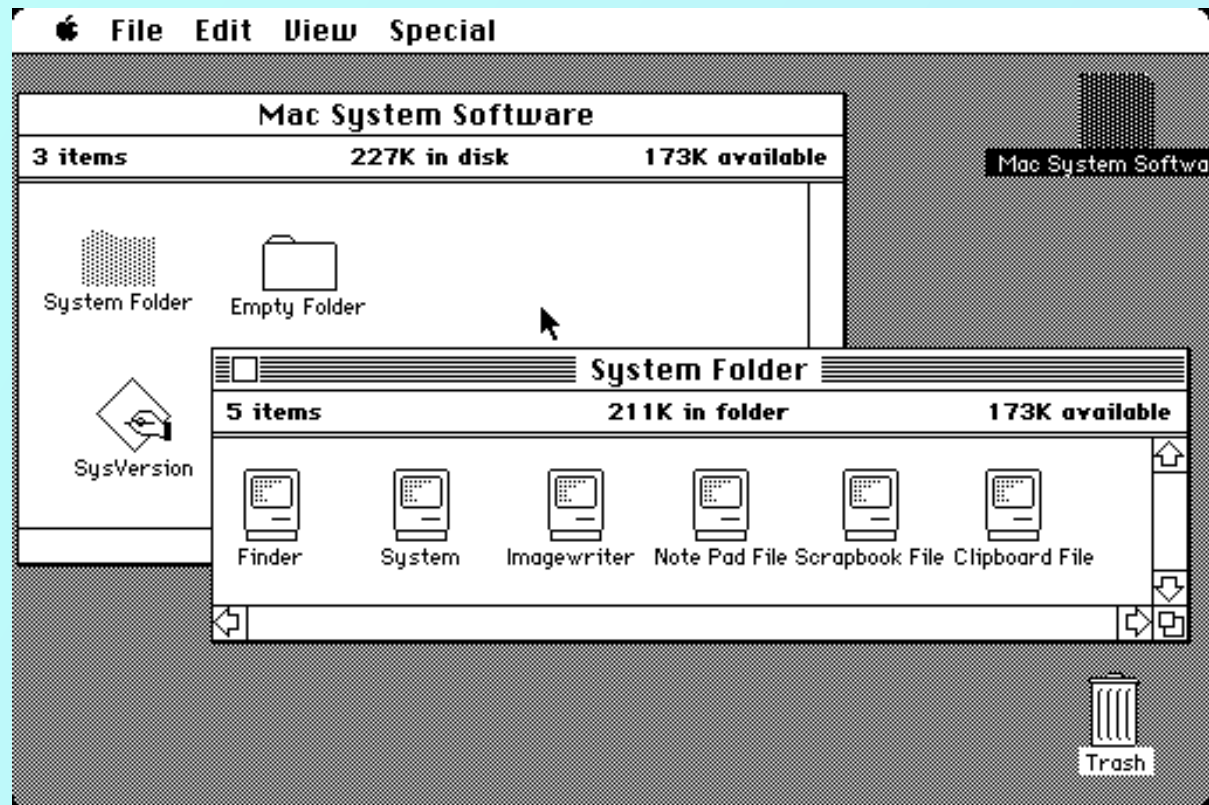(C)1977 MICROSOFT CORP.
ALL RIGHTS RESERVED

READY.

1982 – 1986
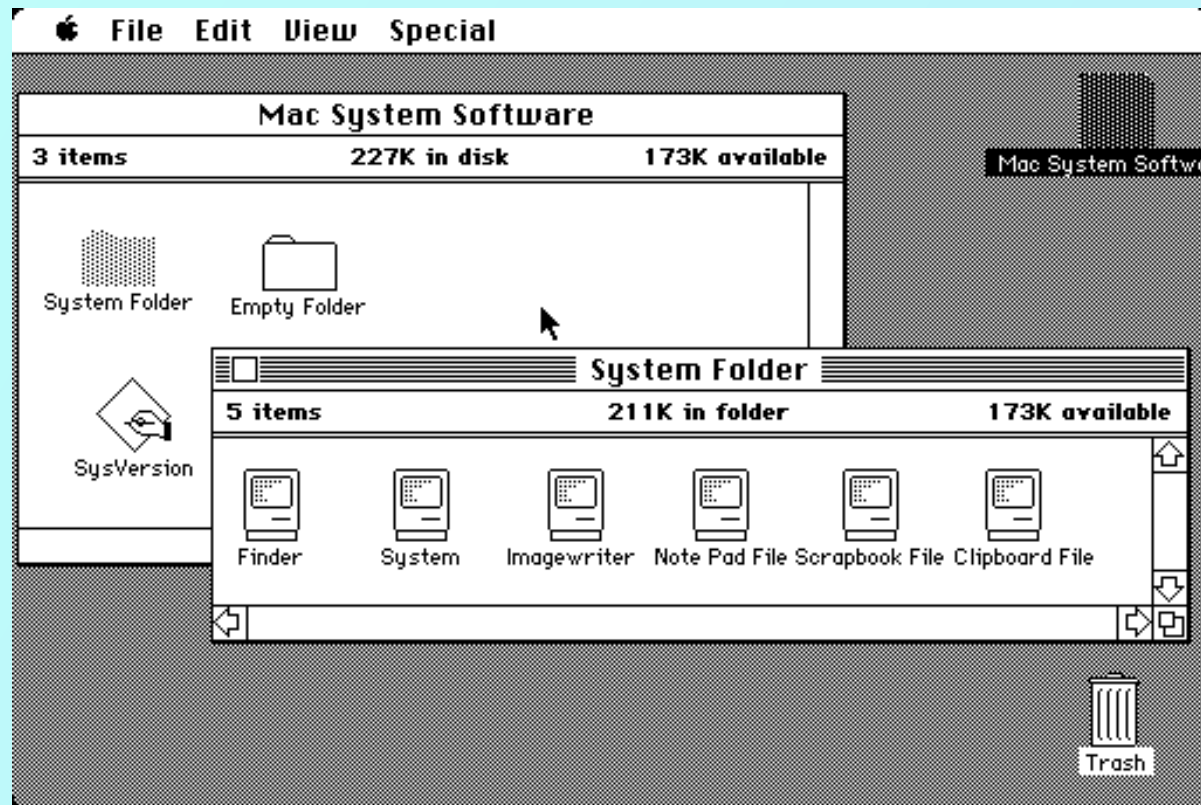
The first assistive (digital) technologies began to emerge

1983 - 1984

Graphical Interfaces

1983 - 1984

Graphical Interfaces === ⛔ Not 1:1 with a11y hardware

1985-2005

A need for standards and regulation emerged

- 1990: The Americans with Disabilities Act (ADA)
- 1994: Birth of the Internet and mainstream adoption of HyperText
- 1995: Windows 95 was the first OS to come with built in accessibility features
- 1996: pwWebSpeak by Productivity Works **first web browser with built in text to speech**
- 1998: Rehabilitation Act Section 508 passed, requiring all government services to be accessible
- 1999: World Wide Web Consortium (W3C) released **Web Content Accessibility Guidelines**.
- 2000: Microsoft Windows 2000 released, with an on screen keyboard option and the ability to translate text to speech for illiterate or blind users (5).
- 2005: OSX 10.4 (Tiger) released with VoiceOver included, though Windows SR was still much better until around 2008

2007-2011

Smart Phones and gesture based interactions

2016 to present

Many companies begin to make a commit toward a11y

Google

- TalkBack
- Voice Access
- Switch Access
- Live Transcribe
- Live Caption
- Lookout

Facebook
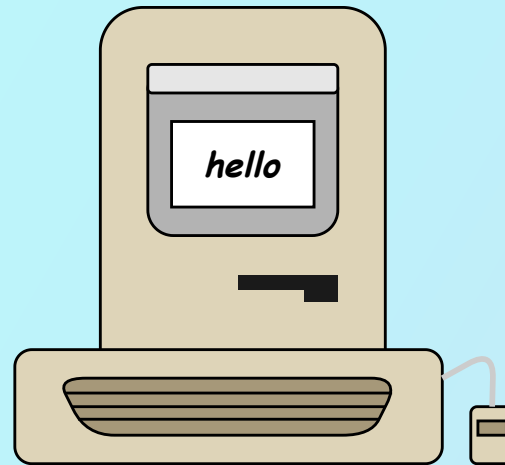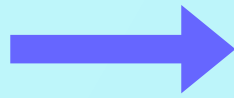
- Automatic alt text

Apple

- VoiceOver built into all products
- Magnifier (with Lidar)
- LiveCaptions iOS 16 (Beta)
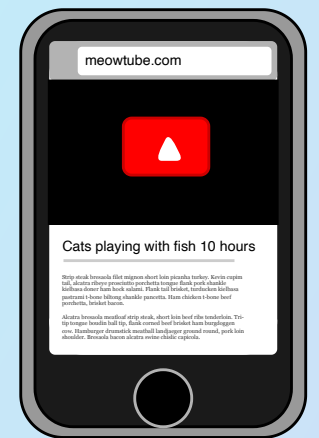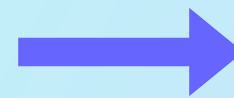- (877) 204-3930

Microsoft

- Live Caption on MS Teams
- Immersive reader

Text   Graphical   Touch

7167 BYTES FREE

READY.

hello

meowtube.com

Cats playing with fish 10 hours

What does it mean to be *accessible*?

It means that your users, *regardless of ability*, can use your product to fulfill their goals

Ability

Experience

Visual Experiences

Blurred

Visual Experiences

Cloudy

Visual Experiences

Ghosting

Visual Experiences

Snowy (static)
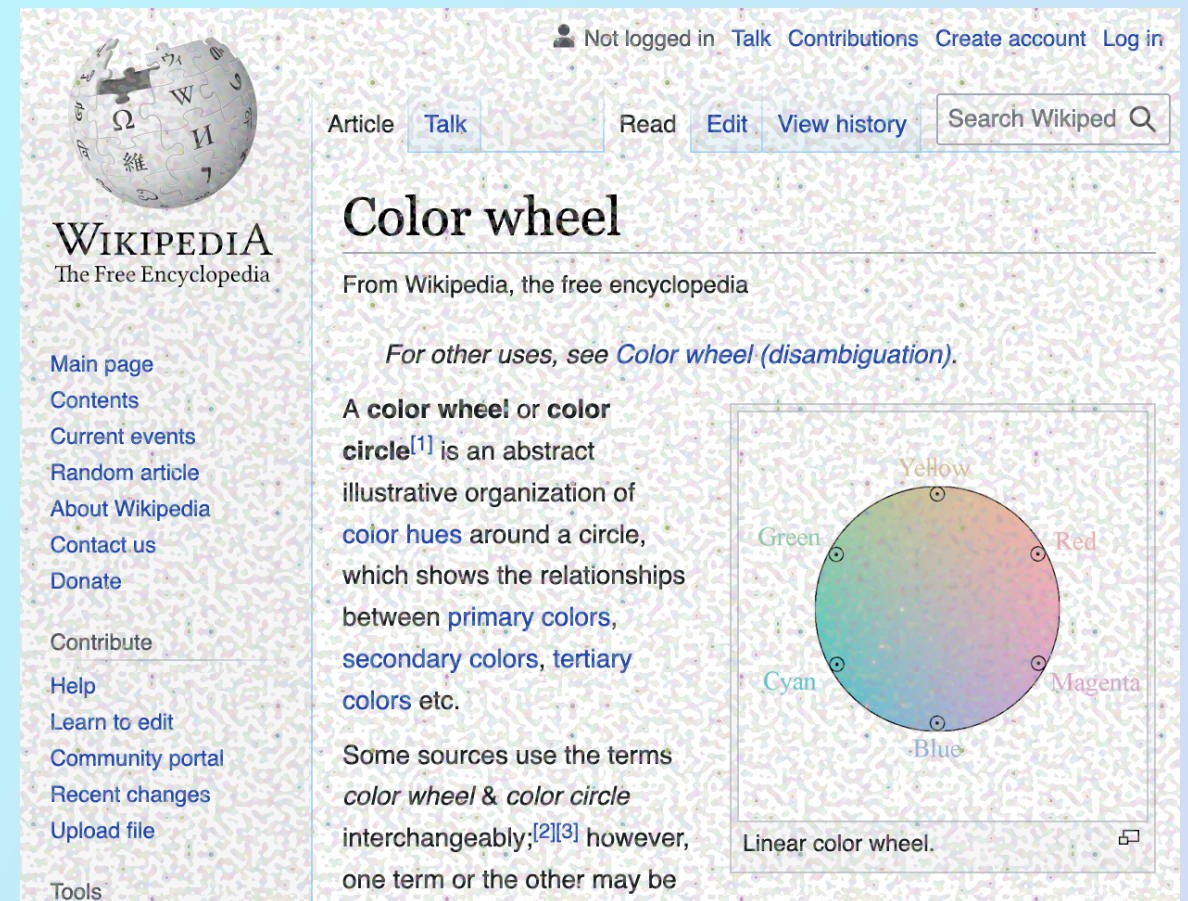
Visual Experiences

Contrast Loss

Visual Experiences

Color loss
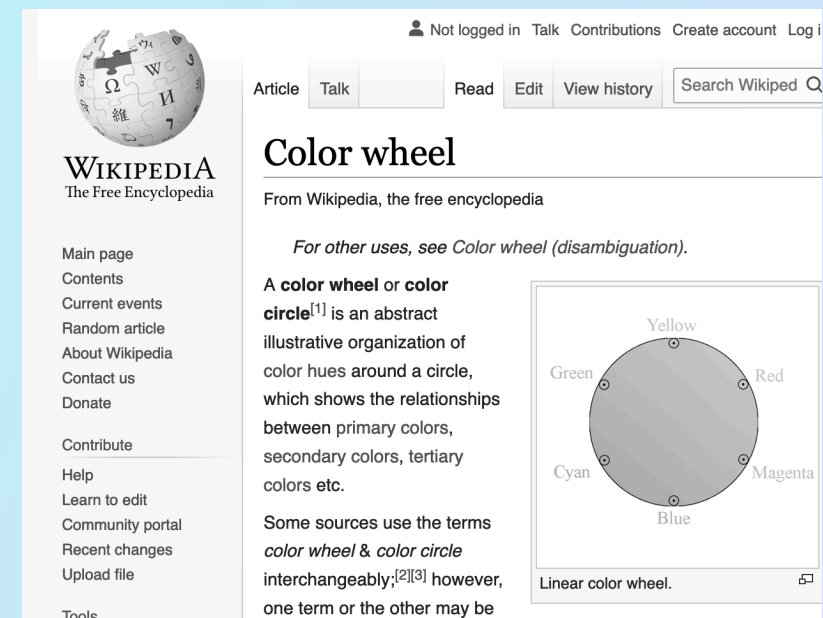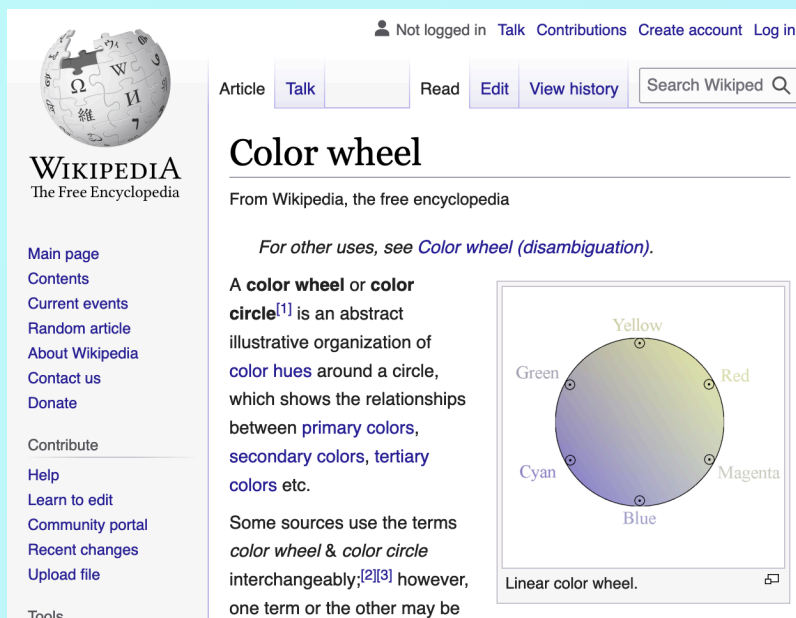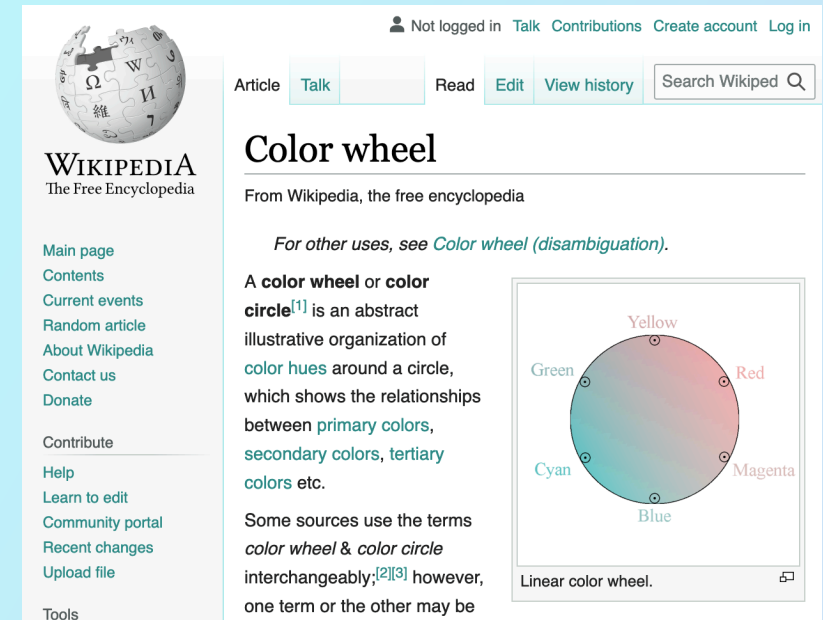
Visual Experiences

Partial Sight Loss
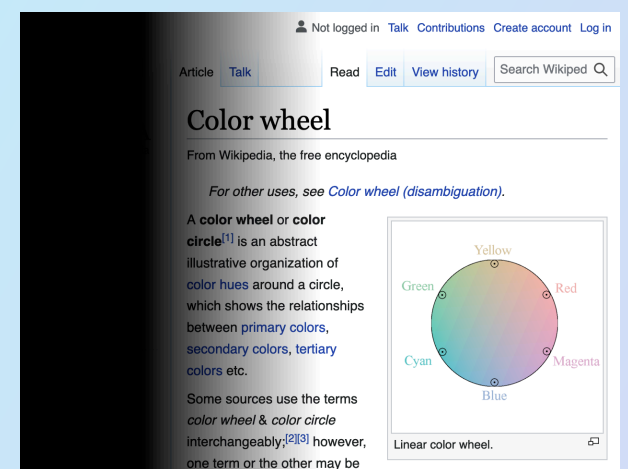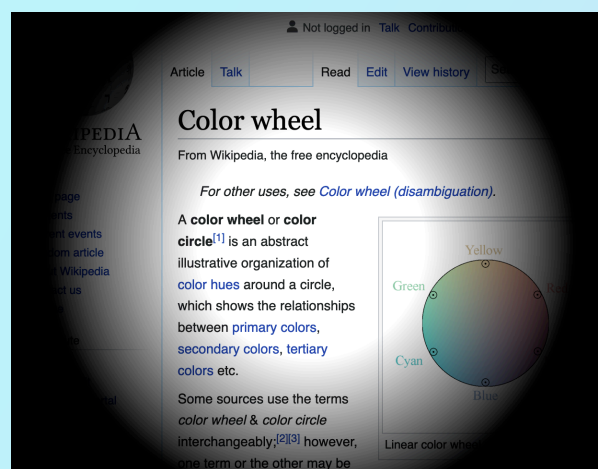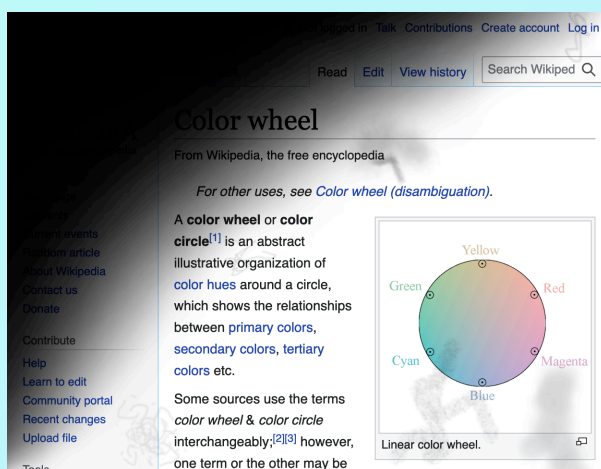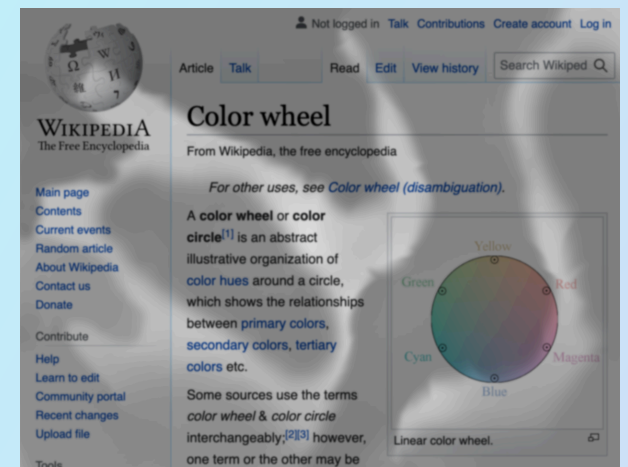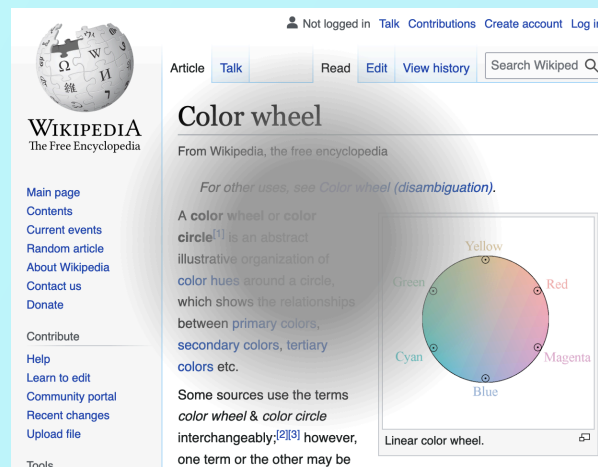
Visual Experiences

Total Sight Loss

Visual Experiences

What should we support as we design and build our apps?

- Contrast and Customizable Colors

- Screen Magnifiers and Zoom

- Screen Reader Compatible
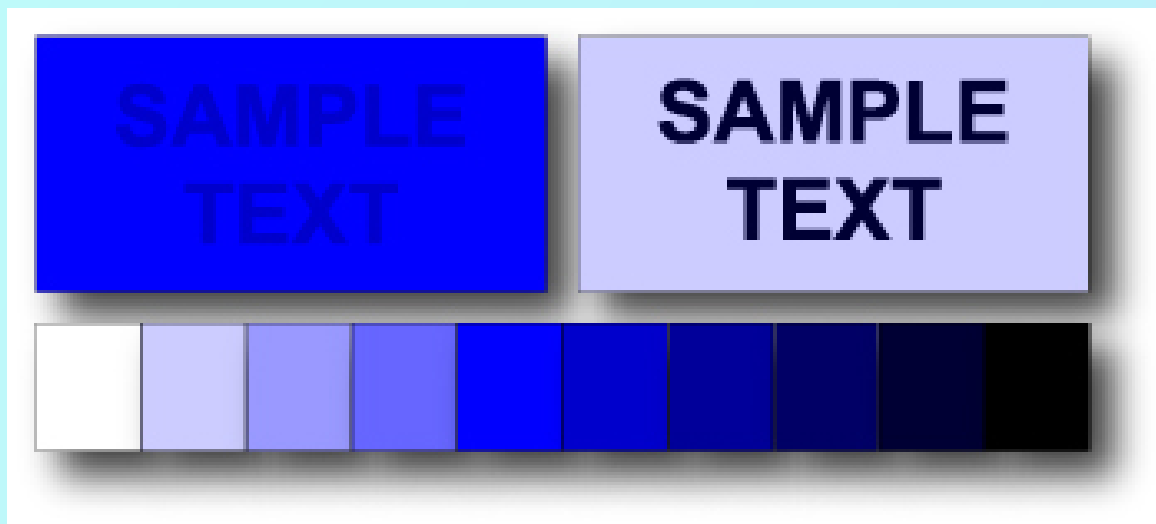
Auditory Experiences
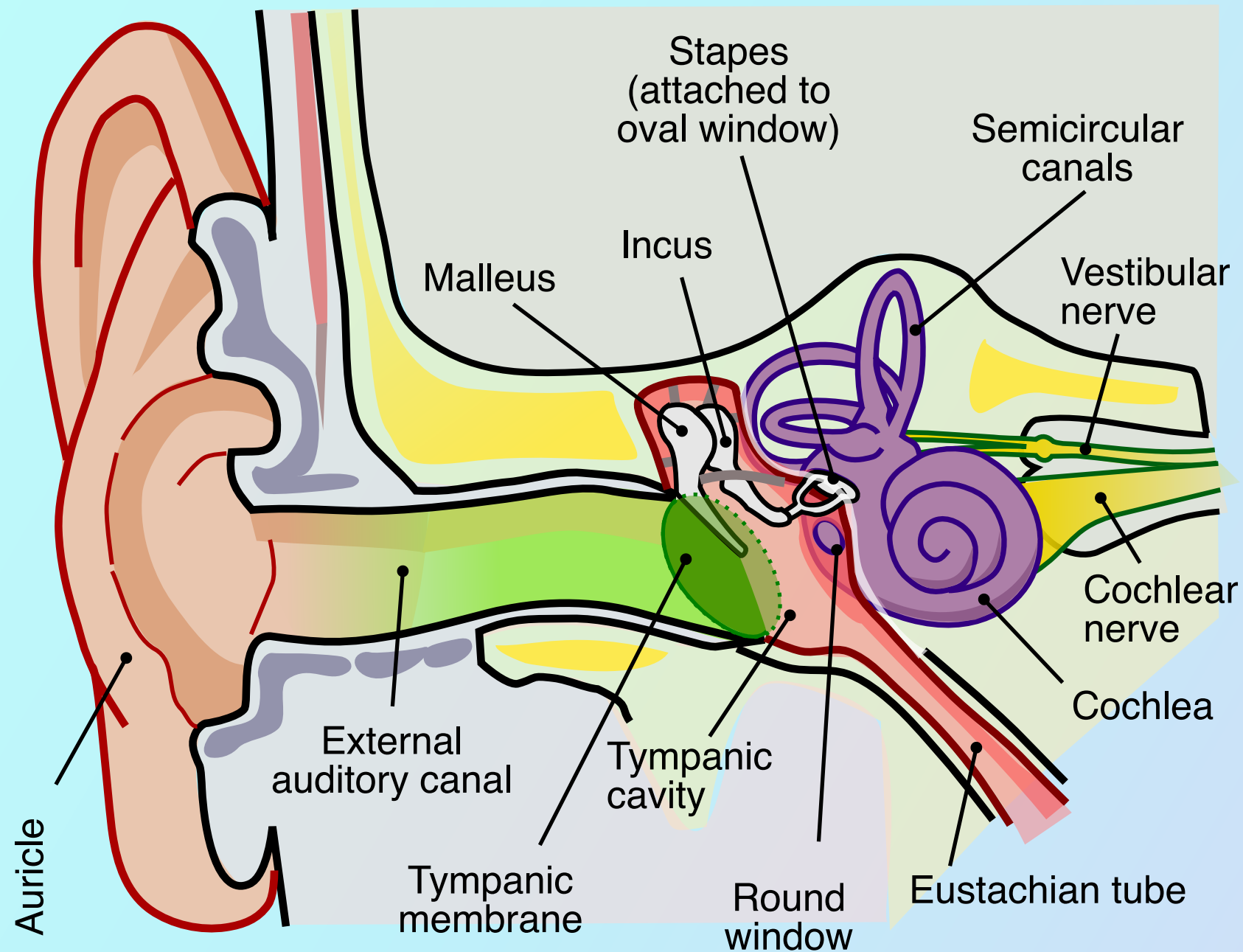
Motor Experiences

Your users might...

- avoid using the mouse
- find it hard to use a typical keyboard
- use voice-controlled software instead of physical interaction
- grow fatigued while using technology
- may not physically interact with their devices directly

Not only do I type with a headstick, but I drive my power wheelchair with my face. Having a strong neck is crucial to my daily function, particularly as a blogger.

— Debbielynne Kespert, Christian blogger

Motor Experiences

Technologies Used

- Special keyboard - 20%
- Special mouse-like device - 33%
- Mouth or head wand - 7%
- Switch device (e.g., puff-sip switch) - 2%
- Eye or head tracking - 2%
- Speech recognition - 30%
- Software to facilitate text entry - 17%
- Virtual or on-screen keyboard - 11%
- Screen reader - 13%
- Other - 22%
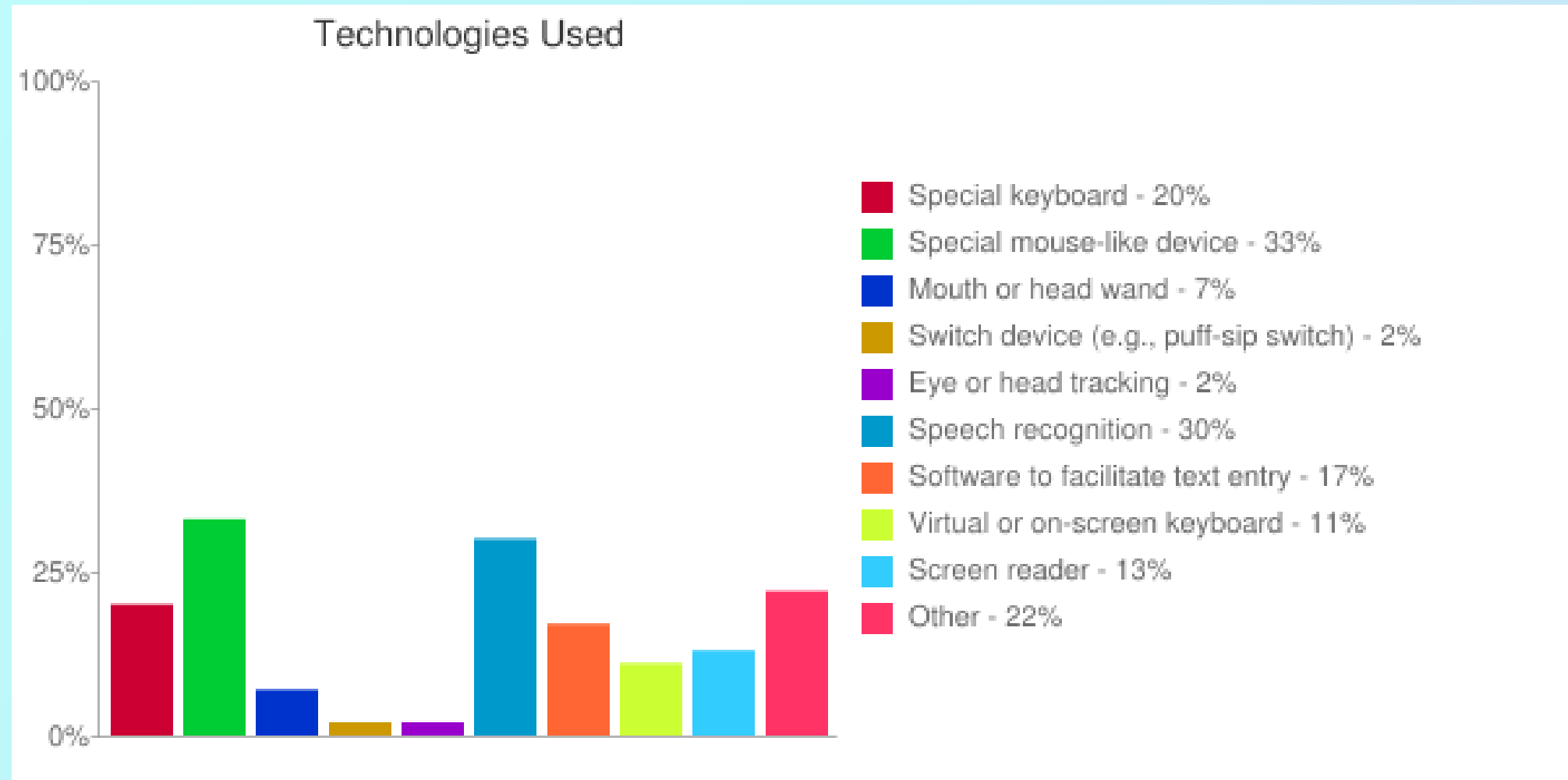
Other Experiences

Physical Challenges:

- Photosensitivity
- Vestibular (inner ear) sensitivity

Cognitive Challenges:

- Memory limitations
- Attention
- Problem-solving
- Comprehension:
  - Visual
  - Reading, linguistic, and verbal
  - Math

The Mighty Screen Reader

Primary Screen Reader
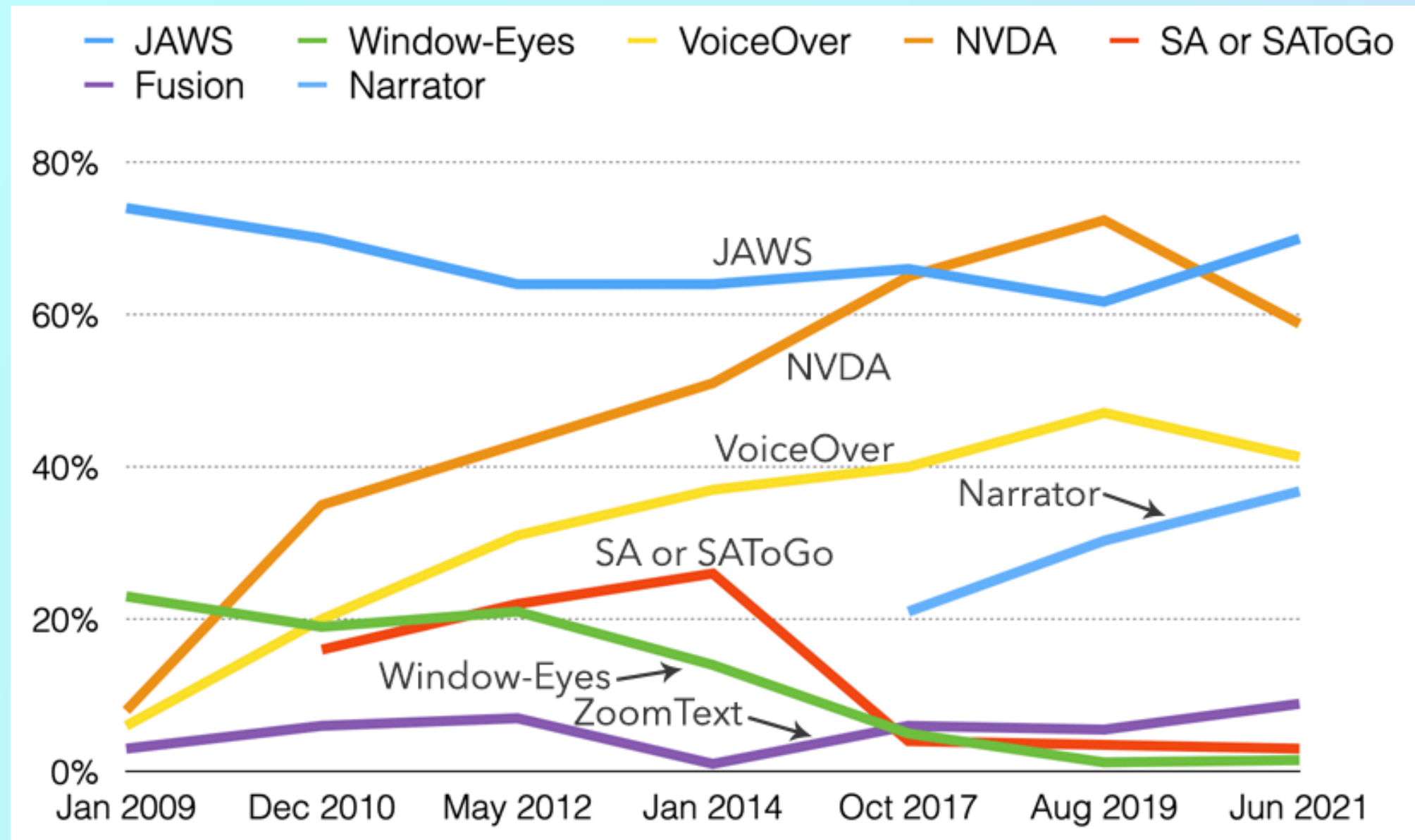
Most SR users own Windows devices

Browser (Desktop)
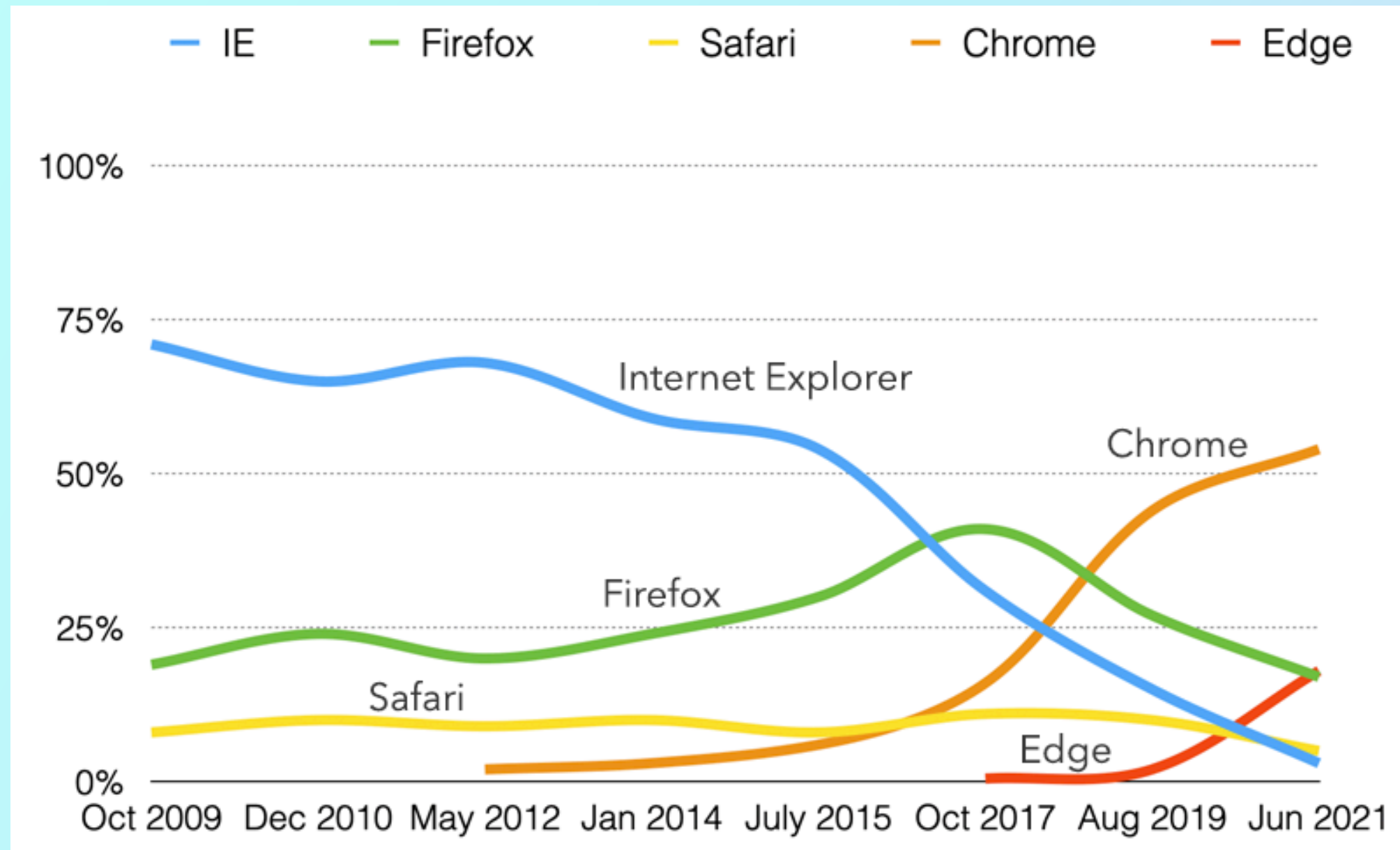
Windows users favor Chromium based browsers

Mobile Screen Reader Usage
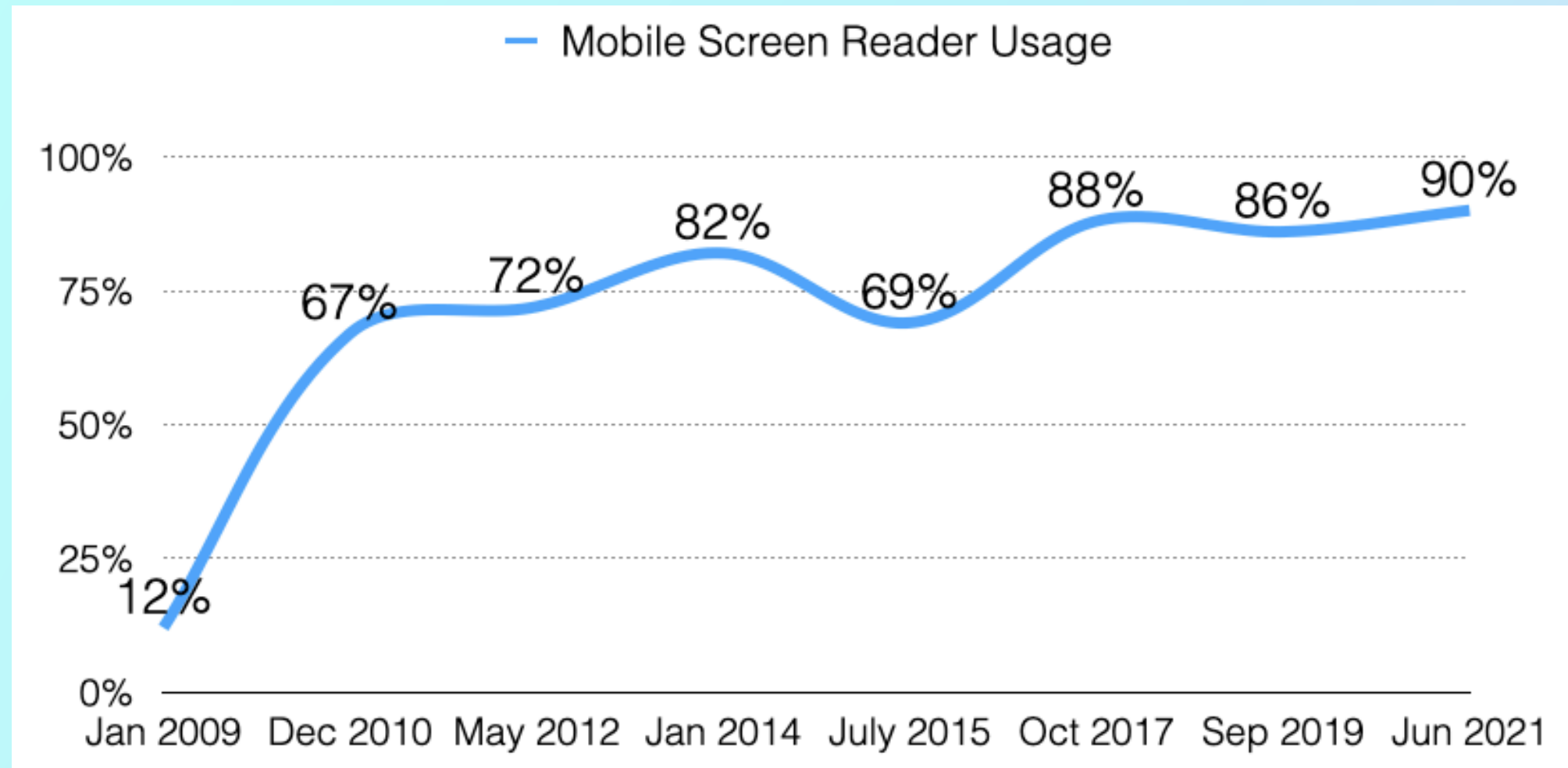
Mobile screen reader usage can not be "ignored"

Mobile OS

Mobile Platform Usage
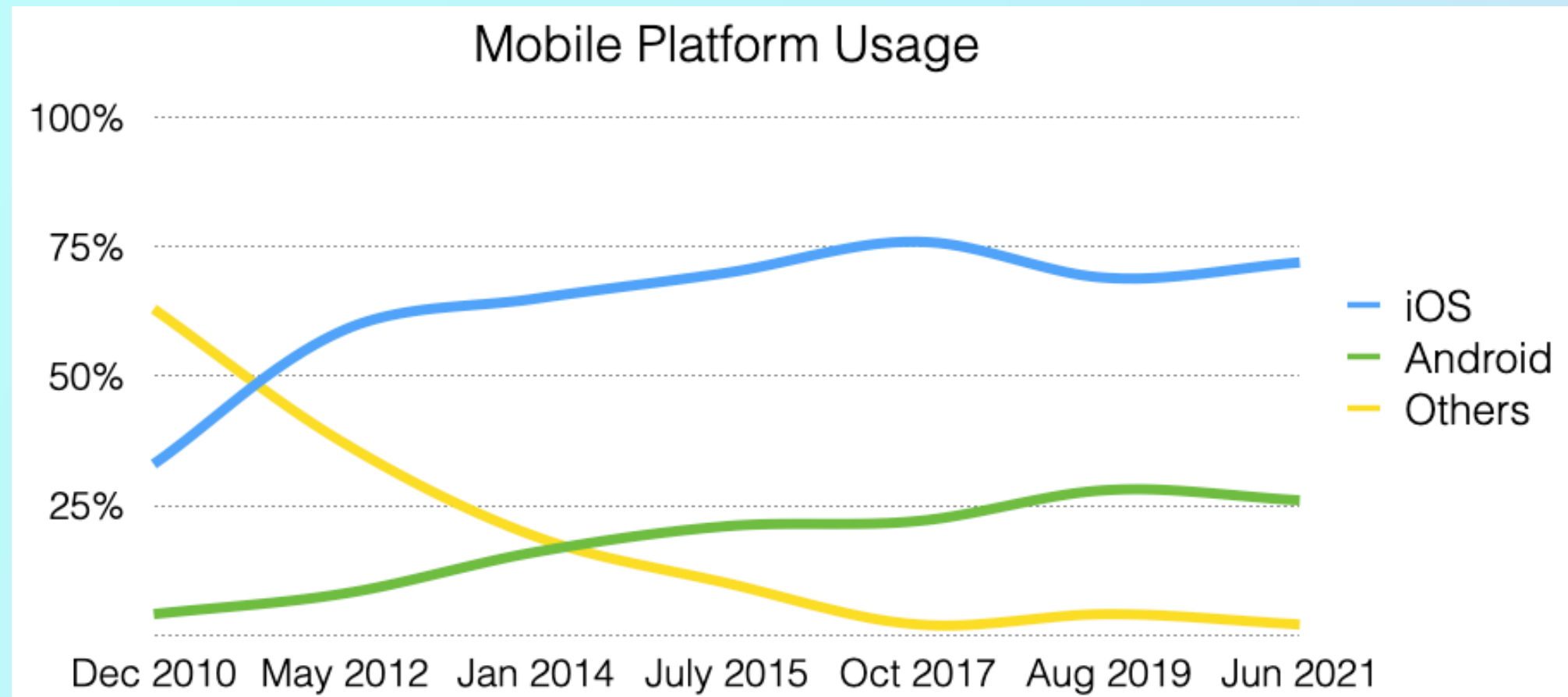
VoiceOver is the screen reader of choice on mobile devices

"My team doesn't have resources to cover all of these. What should I do?"

- Pick a screen reader, and know it well
- Audit your product
- Become familiar with current standards and practices

"Which is the *best* screen reader for me to start with?"

- If your application is desktop focused: JAWS (Windows)
- If your application is mobile-first: VoiceOver (Mac)

Standards, Practices & Patterns

Standards, Practices & Patterns

- Web Content Accessibility Guidelines (WCAG)

- Accessible Rich Internet Applications (ARIA)

- Web Accessibility in Mind (WebAIM)

Contrast Checker

https://webaim.org/resources/contrastchecker/

Let's talk about the DOM

HTML

```
<h1>Hello, World!</h1>
```

JavaScript

```
elem.createElement('img')
```

CSS

```
display: grid;
```

Document Object Model

© 2022 Kyle West

Let's talk about the DOM

DOM Tree

Painted on Screen

example.com

Hello, World!

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

General Interface

USER

Simplified and ARIA Interpreted

A11y Tree

Native API

A11y Tech

Specialized Interface

<div />

Syntax and Semantics

Syntax (or *structure*) comes from the language

```
<tag attribute="value">
  <child attribute1="value" attribute2="value" />
  Content
</tag>
```

Semantics (or *meaning*) comes from the vocabulary we choose

```
<button onclick="openSettingsDialog">
  <img alt="Gear Icon" src="./icons/gear.svg" />
  Settings
</button>
```

Everybody loves `div`

```html
<div class="sidebar">
  <div class="option-group">
    <div class="bold">Price</div>
    <div class="option">
      <input type="radio" name="price" value="-25"/>
      Up to $25<br/>
    </div>
    <div class="option">
      <input type="radio" name="price" value="25-100"/>
      $25 to $100<br/>
    </div>
    <div class="option">
      <input type="radio" name="price" value="100+"/>
      $100 & above<br/>
    </div>
  </div>
  <!-- ... [other filter options] ... -->
  <button type="submit">Search</button>
</div>
```

Everybody should love *all* HTML

```html
<aside>
  <form>
    <fieldset>
      <legend>Price</legend>
      <input type="radio" id="below25" name="price" value="-25"/>
      <label for="below25">Up to $25</label><br/>

      <input type="radio" id="below100" name="price" value="25-100"/>
      <label for="below100">$25 to $100</label><br/>

      <input type="radio" id="over100" name="price" value="200+"/>
      <label for="over100">$100 & above</label><br/>
    </fieldset>
    <!-- ... [other filter options] ... -->
    <button type="submit">Search</button>
  </form>
</aside>
```
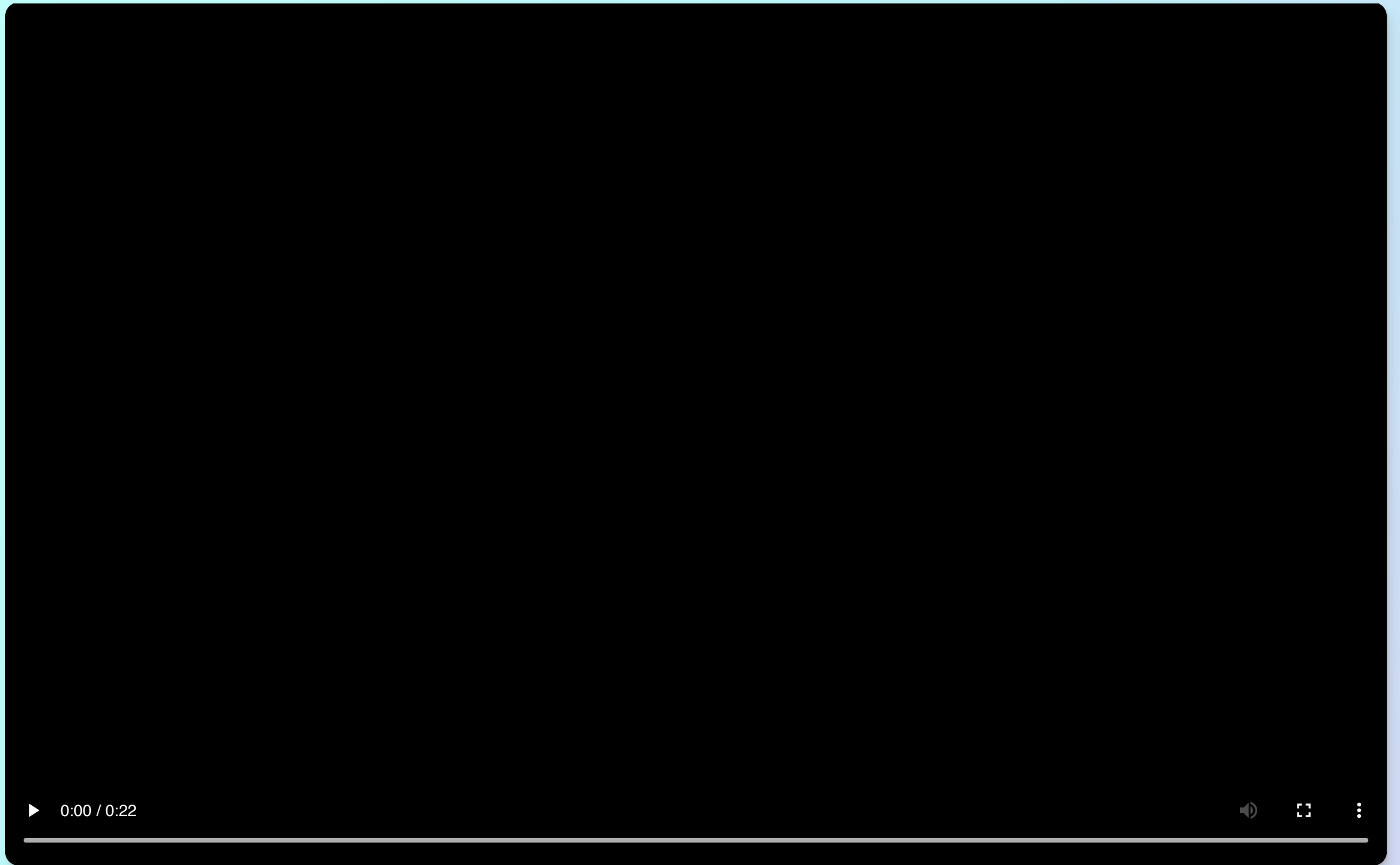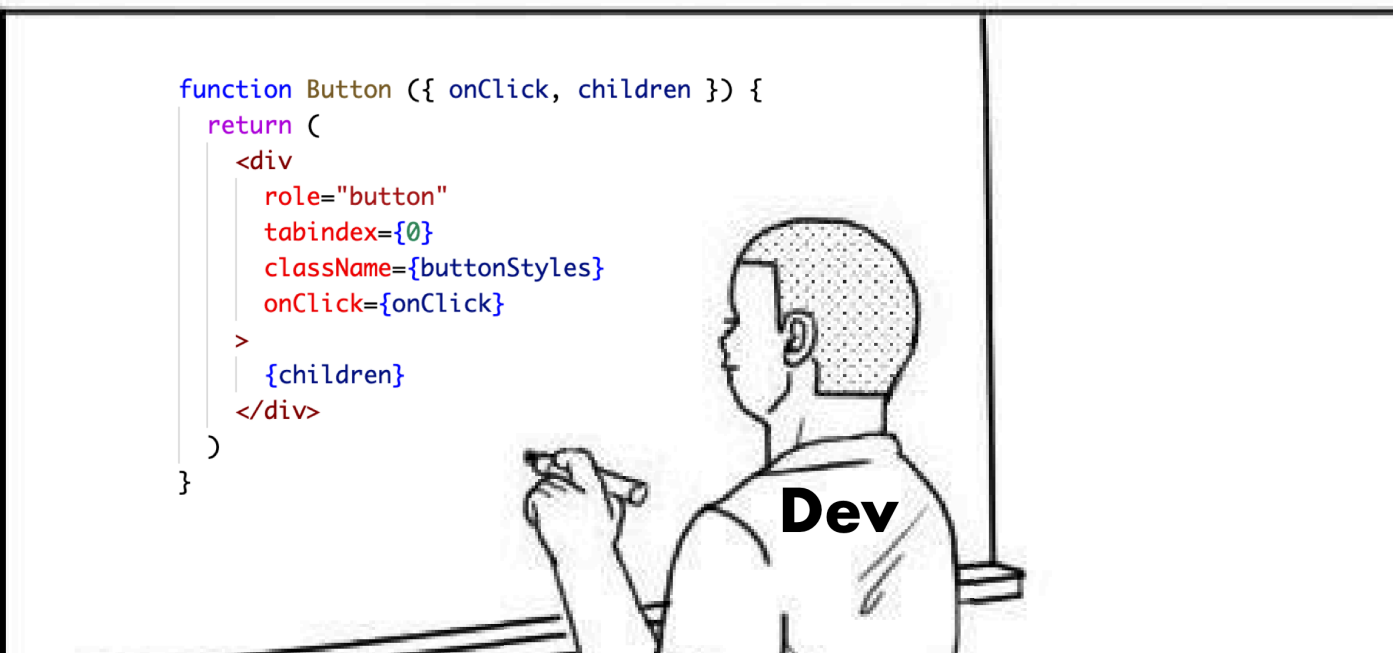
Naive HTML (Try it out: kylewest.dev/cp/rNvqVZW)

Semantic HTML (Try it out: kylewest.dev/cp/rNvqVZW)

© 2022 Kyle West

He should have just used a native `<button>`
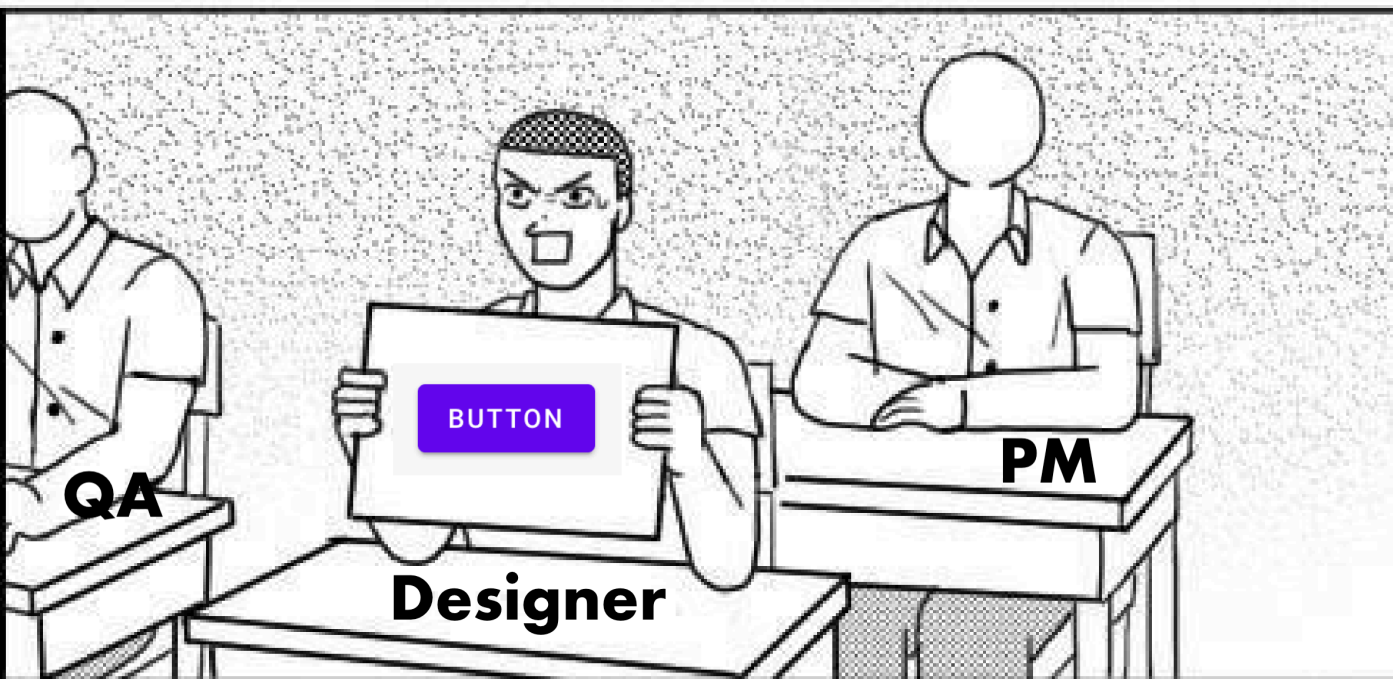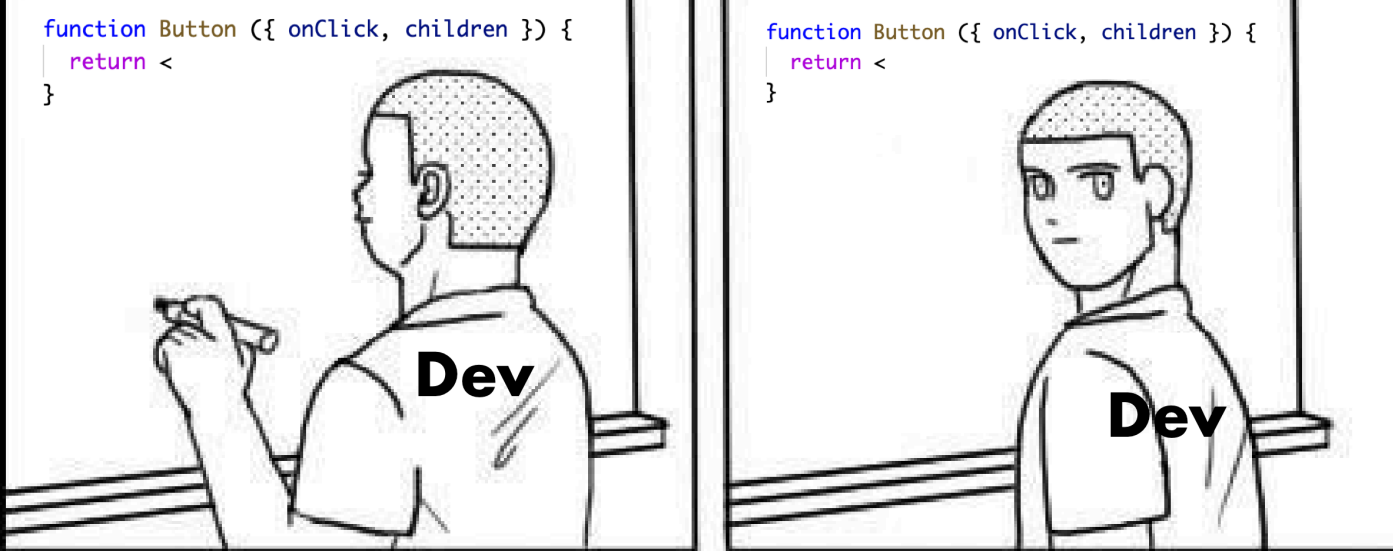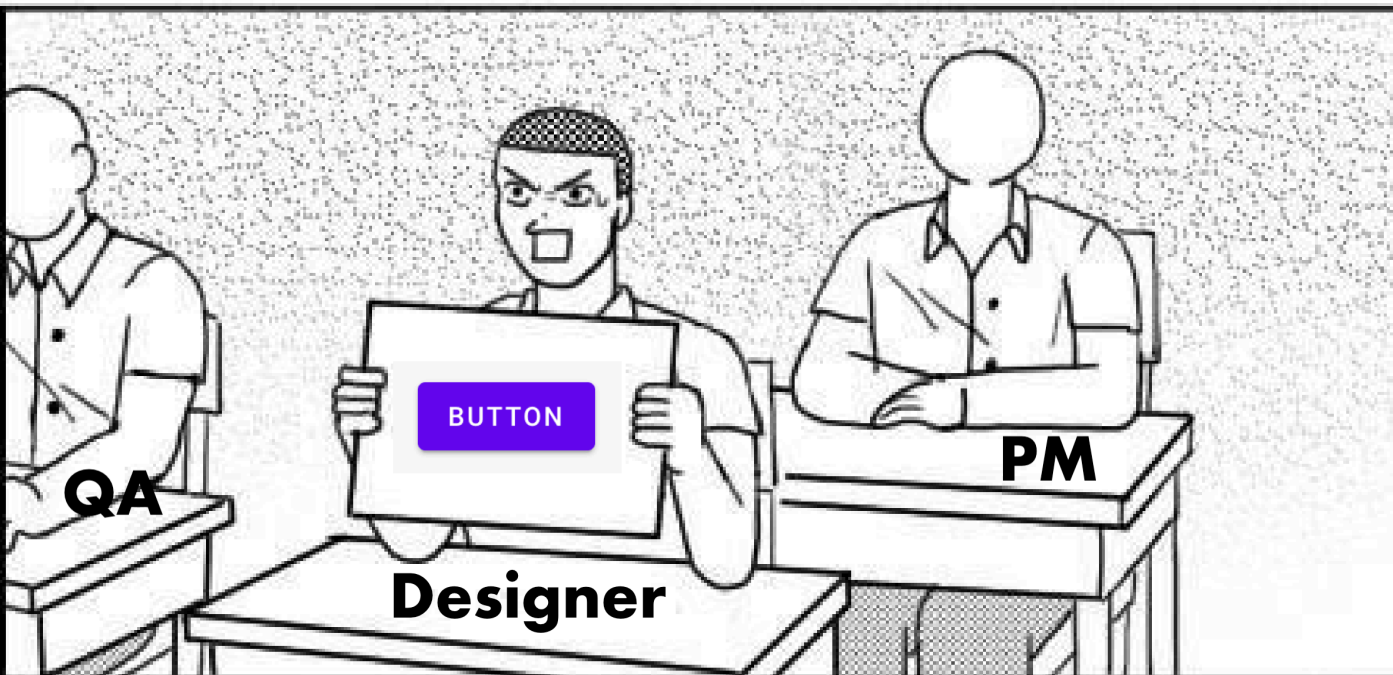
```
function Button ({ onClick, children }) {
  return (
    <button
      className={buttonStyles}
      onClick={onClick}
    >
      {children}
    </button>
  )
}
```

His solution works, but it is incomplete
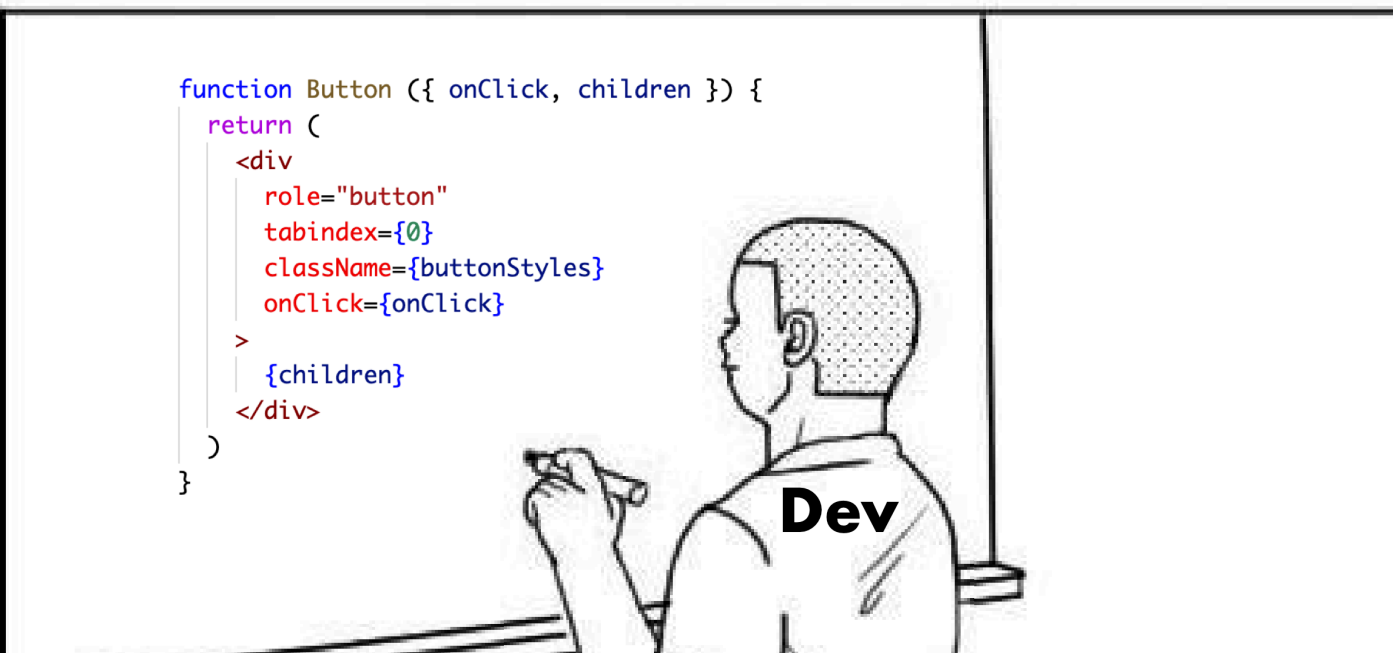
No ARIA is better than Bad ARIA

" Functionally, ARIA roles, states, and properties are analogous to **CSS for assistive technologies**. For screen reader users, ARIA controls the rendering of their non-visual experience.

Incorrect ARIA misrepresents visual experiences, with potentially devastating effects on their corresponding non-visual experiences. "

— ARIA Authoring Practices Guide
(emphasis added)

role

as in

```
<div role="feed">
```

Two important principles

1. A role is a promise to the screen reader

```html
<div role="button">Click Me</div>
```

2. ARIA can both cloak & enhance, creating both power & danger

```html
<ul role="navigation">
  <li><a href="uri1">nav link 1</a></li>
  <li><a href="uri2">nav link 2</a></li>
</ul>
```

```html
<nav>
  <ul>
    <li><a href="uri1">nav link 1</a></li>
    <li><a href="uri2">nav link 2</a></li>
  </ul>
</nav>
```

The accessibility tree will error on this:

`ERROR! Previous list items are not in a list!`

Expressing semantics

The following are equivalents
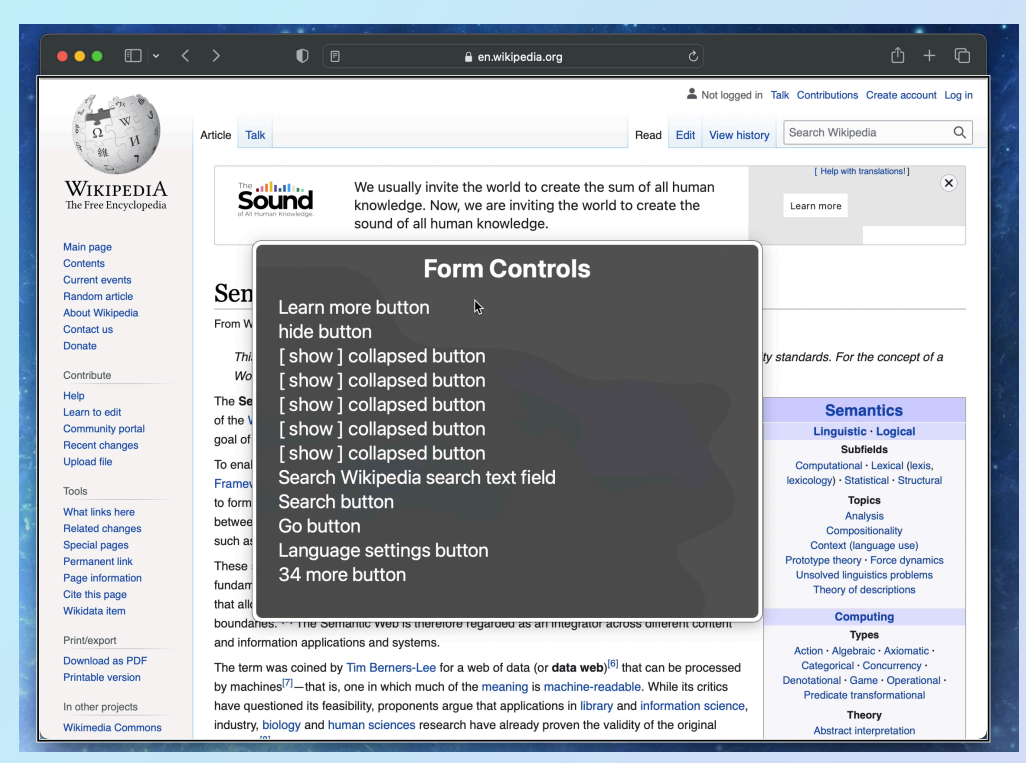
```
<table>

<div role="table">

<button role="table">
```

Even CSS effects the Accessibility Tree

```
<style>
  .table {
    display: table;
  }
</style>

<button class="table">
```

When in doubt, `role` always wins

https://codepen.io/kylewestrrr/pen/abWMWZX

Landmarks

© 2022 Kyle West

Traversable Semantics

© 2022 Kyle West
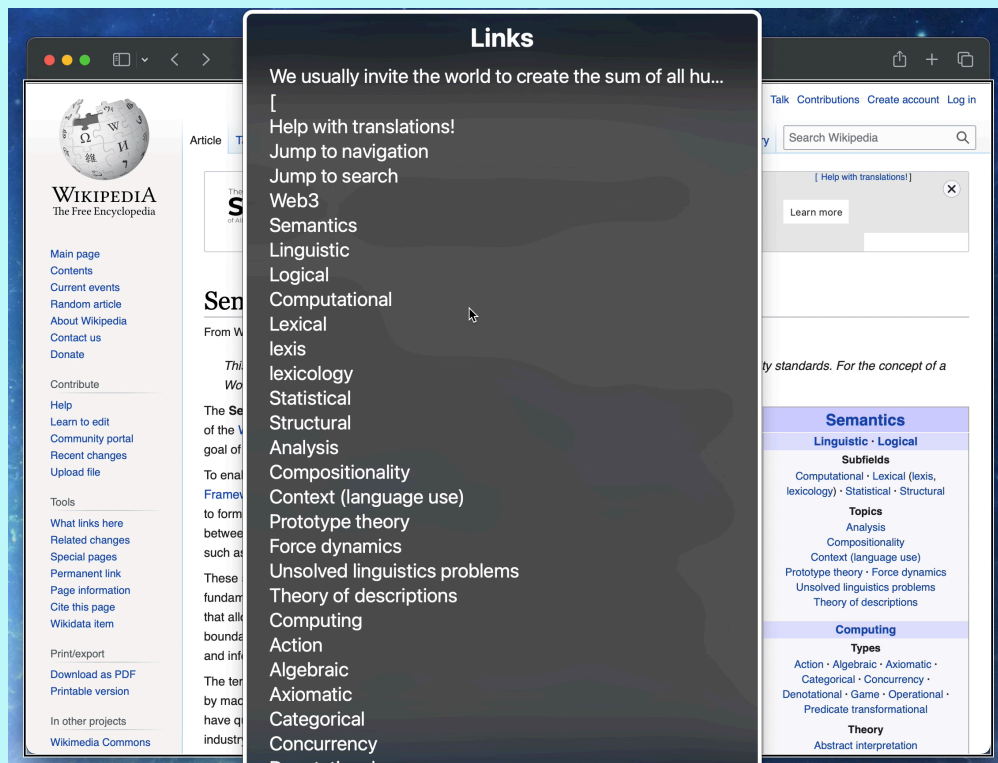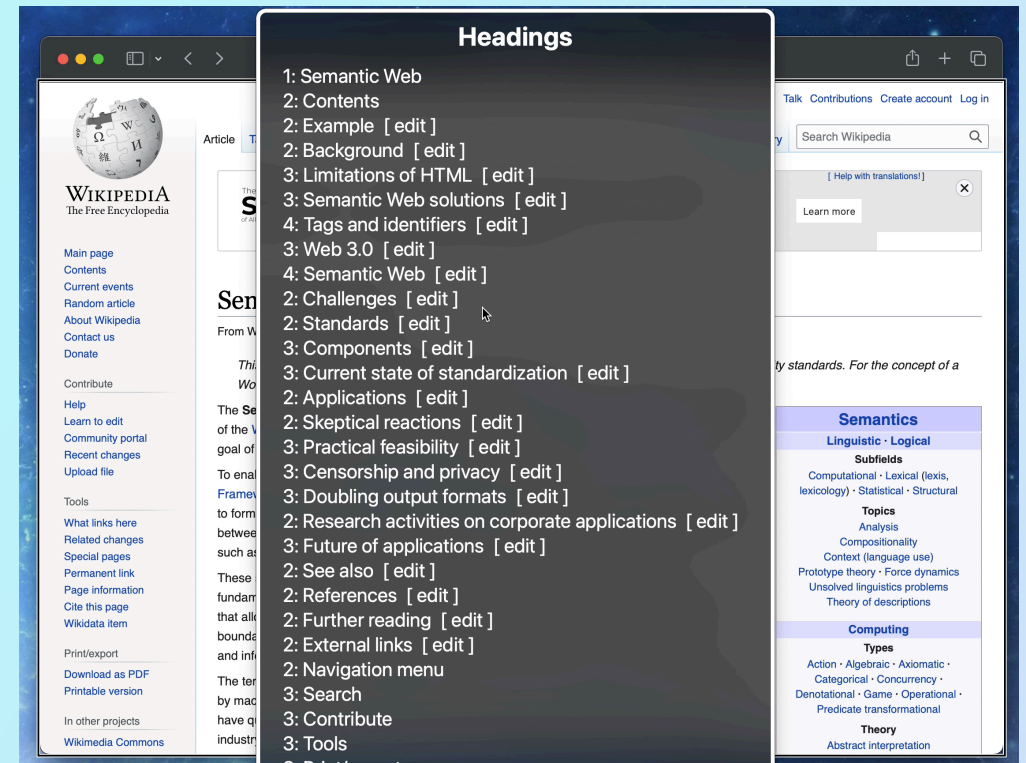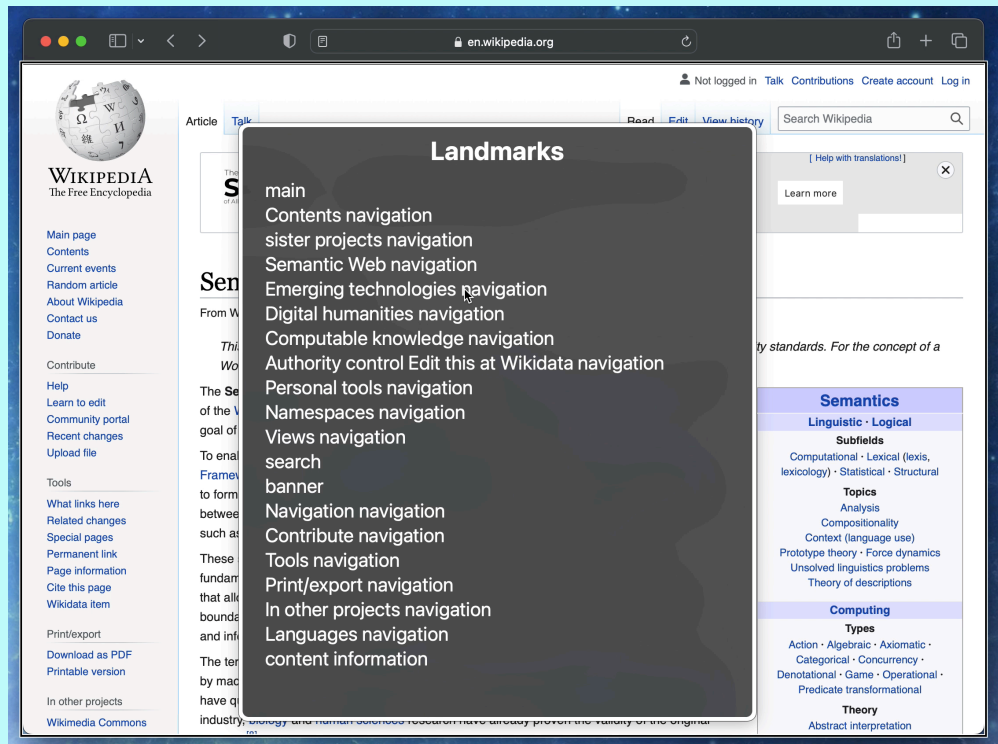
Naming and describing content

- name with child content

- name with a string attribute via `aria-label`

- name by referencing content with `aria-labelledby`

- name form controls with the `<label>` element

- name `<fieldset>` with the `<legend>` element

- name or describe `<table>` and `<figure>` with `<caption>`

- fallback names are derived from `title` and `placeholder` attributes

- describe images by providing an `alt` attribute

- describe content by referencing content with `aria-describedby`

When to hide

Do this (when it makes semantic sense):

```
<img role="presentation" />

<svg aria-hidden>
  <!-- ... -->
</svg>
```

DO NOT do this:

```
<button aria-hidden>Settings</button>
```

What's wrong with this?

Here is some real code I once saw:

```html
<body>
  <div id="root" aria-hidden></div>
  <div id="screen-reader-root" class="visually-hidden"></div>
</body>
```

Resources

ARIA Authoring Practices Guide (APG)

https://www.w3.org/WAI/ARIA/apg/patterns/

Web Accessibility in Mind (WebAIM)

https://webaim.org

Web Content Accessibility Guidelines (WCAG)

https://www.w3.org/WAI/standards-guidelines/wcag/

Recap

- WCAG is the legal standard for Designing

- ARIA is the applicable standard for Developing

- WebAIM is a great resource for learning about real user experiences

- Make sure you have correct heading levels

- Label your inputs and images

- You should probably learn to a screen reader

Accessibility

=

Usability

Kyle West

Slides can be found at

kylewest.dev/slides

Feel free to reach out on Twitter with questions

@KyleWestCS

Attribution

In order of appearance