

Accessibility by Design:

Practical A11y for Everyday Software Engineering

That's me

- Software Engineer
 - Elixir/Phoenix,
 - Postgres,
 - HTML, CSS, JS
- Conferences
- Meetup Organiser
- Team Lead and Project Lead



... also me

- Family Friends, the pack:
 - 5 cats
 - 3 dogs
 - 2 pigs
- Lives in the middle of nowhere in Franconia (Bavaria)



*The goal is not to make everyone an accessibility expert,
but to make accessibility a natural part
of our everyday engineering work.*

01

The What

02

The Why

03

The How

04

A Little Quiz

05

Objections “Best Of”

06

Future Mindset

01

The What

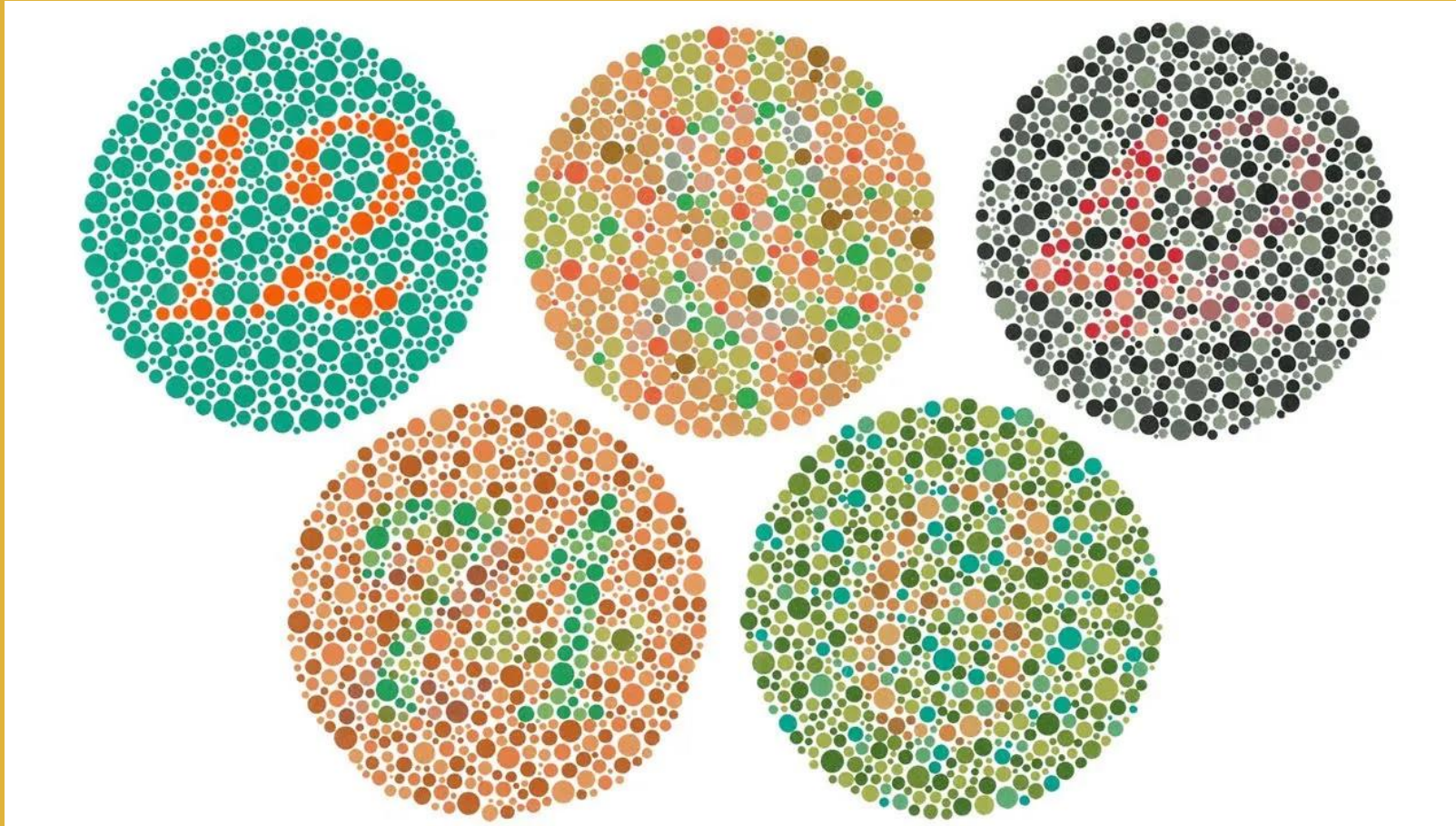
Different (views on) disabilities

Who of you is disabled in some way?

Different (views on) disabilities

	Permanent	Temporary	Situational
 Touch	<i>One arm</i>	<i>Arm injury</i>	<i>Carry stuff</i>
 See	<i>Blind</i>	<i>Cataract</i>	<i>Bright sunlight</i>
 Hear	<i>Deaf</i>	<i>Ear infection</i>	<i>Loud environment</i>
 Speak	<i>Non-verbal</i>	<i>Laryngitis</i>	<i>Heavy accent</i>
 Think	<i>Cognitive disability</i>	<i>Fatigue</i>	<i>Distracted by multi-tasking</i>

Can you read the numbers ... easily?



Source: Derek Torsani, <https://medium.com/@dmtors/being-a-color-blind-designer-aa6828c09a42>

Have you ever ...

*... tried using your phone one-handed
while holding a baby?*

Have you ever ...





... struggled to read a screen because of bright sunlight?

Have you ever ...

... used subtitles on a video?

Who of you is disabled in some way?

Different (views on) disabilities: Web Development

	Permanent	Temporary	Situational
 Touch	<i>One arm</i>	<i>Arm injury</i>	<i>Carry stuff</i>
 See	<i>Blind</i>	<i>Cataract</i>	<i>Bright sunlight</i>
 Hear	<i>Deaf</i>	<i>Ear infection</i>	<i>Loud environment</i>
 Connect	<i>Low bandwidth</i>	<i>Data cap reached</i>	<i>On a train/tunnel</i>

02

The Why

Empathy, Common sense ... & the EAA

For whom is that feature useful?



... all of us 😊

Source: Nick-Philly, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=84618901>

Why are we dealing with A11y?

01

For Others

To include the largest number of people possible to experience your app.

02

For Us

... that includes all of us.

03

For Compliance

To satisfy the European Accessibility Act.

European Accessibility Act — EAA

- When
 - In force since 28 June **2025** for new products
 - Legacy products must comply by 28 June **2030**
- Who
 - Targets consumer-facing (**B2C**) products and services sold **to consumers in the EU**
- What
 - Compliance with EU standard EN 301 549, which in practice means **WCAG 2.1 AA**

Web Content Accessibility Guidelines— WCAG

- International standard for web accessibility by the World Wide Web Consortium (W3C).

Quick Aside

POUR Principles

PERCEIVABLE

P Users must be able to see or hear the content

OPERABLE

O Users must be able to interact with the interface

UNDERSTANDABLE

U Content and operation must be understandable

ROBUST

R Content must be interpreted by assistive technologies

Web Content Accessibility Guidelines— WCAG

- International standard for web accessibility by the W3C.

A

AA

AAA

- WCAG A, AA, AAA structured checklist.

03

The How

Assistive tech, Accessibility Tree, Native Elements & ARIA,
Tooling & Testing, PR Review Checklist

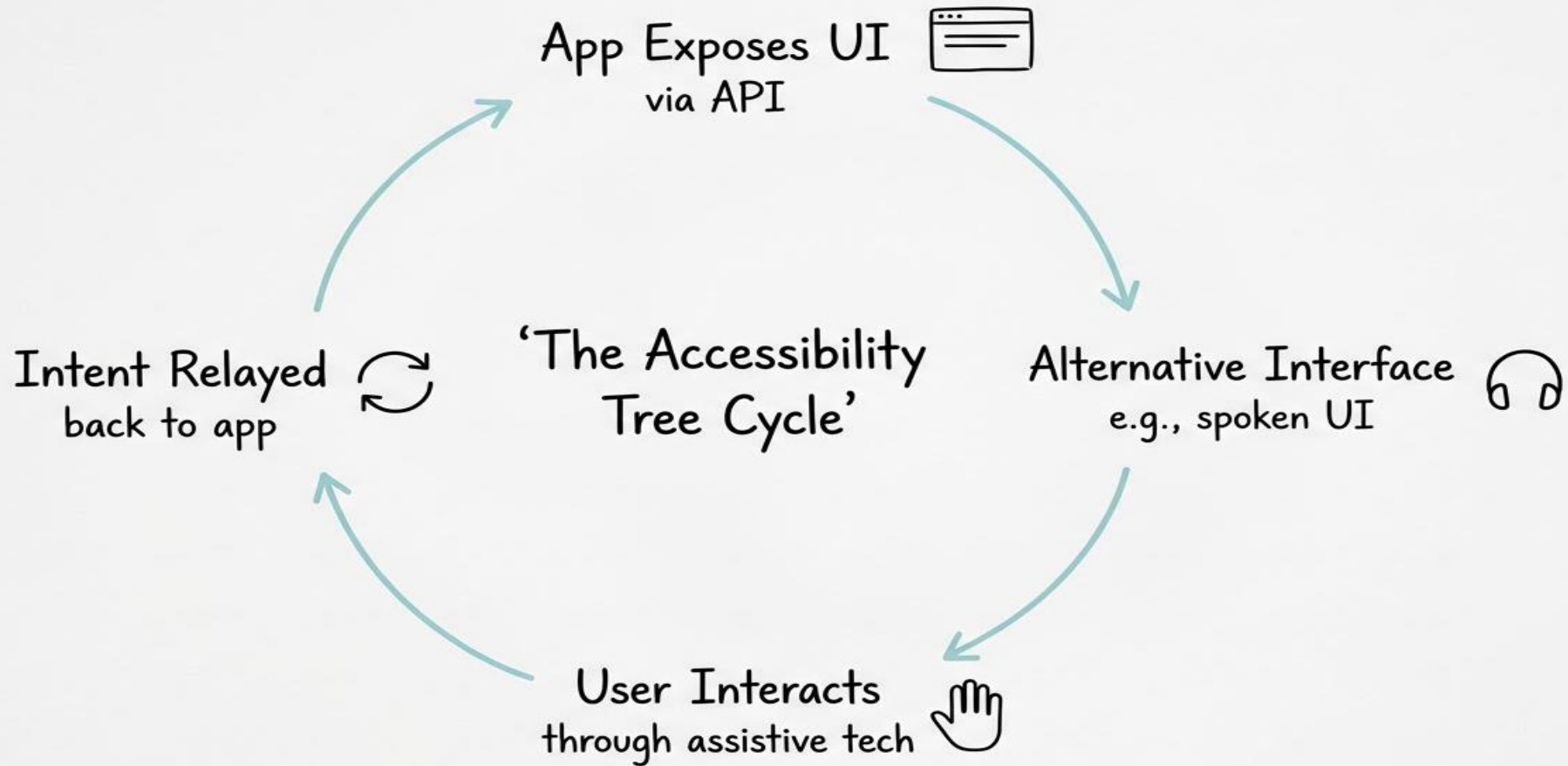
Assistive Tech for the Web

- Computer Access Tools
- Visual Aids and Magnification Systems
- Screen Readers
 - Text-to-Speech
 - They identify form fields
 - Interpret image alternative text
 - They navigate linearly or jump between elements like *headings*, *links*, and *landmarks*

The Accessibility Tree

- Let's take a look what the screen reader sees!

The Accessibility Tree Cycle



The Web: HTML, CSS, JS

- HTML = nouns
- CSS = adjectives
- JS = verbs

Most accessibility wins are in the nouns.



```
1 <div>
2   <span>Three words</span>
3   <div>
4     <a>one word</a>
5     <a>one word</a>
6     <a>one word</a>
7     <a>one word</a>
8   </div>
9 </div>
10 <div>
11   <div>
12     <div>five words</div>
13   </div>
14   <div>
15     <div>three words</div>
16     <div>forty-six words</div>
17     <div>forty-four words</div>
18   </div>
19   <div>
20     <div>seven words</div>
21     <div>sixty-eight words</div>
22     <div>forty-four words</div>
23   </div>
24 </div>
25 <div>
26   <span>five words</span>
27 </div>
```


```
1 <header>
2   <h1>Three words</h1>
3   <nav>
4     <a>one word</a>
5     <a>one word</a>
6     <a>one word</a>
7     <a>one word</a>
8   </nav>
9 </header>
10 <main>
11   <header>
12     <h2>five words</h2>
13   </header>
14   <section>
15     <h2>three words</h2>
16     <p>forty-six words</p>
17     <p>forty-four words</p>
18   </section>
19   <section>
20     <h3>seven words</h3>
21     <p>sixty-eight words</p>
22     <p>forty-four words</p>
23   </section>
24 </main>
25 <footer>
26   <p>five words</p>
27 </footer>
```

The Nouns: Native Elements

- MDN: [HTML semantic elements grouped functionally](#)
 - Content sectioning
 - `<header>`, `<main>`, `<footer>`, `<nav>`
 - Text content
 - `<blockquote>`, `<figure>`, `<figcaption>`
 - Demarcating edits
 - ``, `<ins>`
 - Forms
 - `<button>`, `<fieldset>`, `<legend>`, `<label>`
 - Interactive elements
 - `<details>`, `<summary>`

```
1 <!-- Reengineered (functions of a) button -->
2
3 <!-- HTML -->
4 <div id="submit-btn" class="btn" role="button" tabindex="0" aria-disabled="false">Submit</div>
5
6 <script>
7 const btn = document.getElementById('submit-btn');
8
9 // 1. Handle mouse clicks
10 btn.addEventListener('click', handleSubmit);
11
12 // 2. Handle keyboard - Enter key
13 btn.addEventListener('keydown', (e) => {
14 |   if (e.key === 'Enter') handleSubmit();
15 | });
16
17 // 3. Handle keyboard - Space key
18 btn.addEventListener('keyup', (e) => {
19 |   if (e.key === ' ') handleSubmit();
20 | });
21 |
22 // 4. Handle disabled state
23 function disableBtn() {
24 |   btn.setAttribute('aria-disabled', 'true');
25 |   btn.setAttribute('tabindex', '-1');
26 |   btn.removeEventListener('click', handleSubmit);
27 |   // Note: keyboard re-activation still not fully blocked
28 | }
29
30 // 5. Handle form submission
31 function handleSubmit() {
32 |   const form = btn.closest('form');
33 |   if (form) form.requestSubmit();
34 | }
35 </script>
36
37 <!-- CSS - focus ring -->
38 <style>
39 .btn:focus-visible {
40 |   outline: 2px solid #005fcc;
41 |   outline-offset: 2px;
42 | }
```

Let's talk about ARIA: The Golden Rules

- 1** If a native element exists, use it.
- 2** Don't override native semantics unless you absolutely must.
- 3** No ARIA is better than bad ARIA. 

Let's talk about ARIA: "Only Use If" Exceptions

- Multiple landmarks of the same type (e.g., two <nav> elements)
 - ✓ add `aria-label="Primary Navigation"`, ...
- Inputs without visible labels
 - ✓ add `aria-label="Search"`
- Content changes without a page reload (e.g., errors/success, chat messages)
 - ✓ use `aria-live="polite"` or `aria-live="assertive"`.

How to pass the Code Review: Manual Testing

- Keyboard Navigation:
 - **Tab / Tab+Shift:** Move back and forth between interactive elements.
 - **Enter:** Trigger buttons and links.
 - **Space bar:** Trigger buttons, checkboxes, and scroll.
 - **Arrow keys:** Navigate within menus, radio groups, ...
- Dev Tools:
 - Use the emulate vision deficiencies* and color picker.
 - Google Lighthouse: Bad accessibility, Good accessibility

* **CMD+Shift+P -> Show Rendering -> Emulate vision deficiencies**

How to pass the Code Review: Automated Testing

- Automated Testing
 - **Static Analysis:** html-validate, eslint-plugin-jsx-a11y
 - **CI Integration:** Lighthouse CI or axe-core

 Automation only catches ~50% of accessibility issues.

PR Review Checklist

- Is there a native element for this?
- Can I keyboard through my app?
- Does every *non-decorative* icon/image have a text name?
- Does the color contrast pass (→ DevTools)?
- Did Lighthouse/Axe pass in the CI?

04

A little Quiz

Question 1

Which of these HTML elements has zero semantic meaning to a screen reader?

1) `
`

2) `<p>`

3) `<div>`

4) ``

Question 2

If a designer gives you a button that navigates the user to a new page, you should use a `<button>` element.

1) FALSE

2) TRUE

Question 3

You have a decorative icon of a magnifying glass next to a labelled search input.

What should the alt attribute look like?

1) alt="Search"

2) alt="Magnifying glass"

3) alt=""

4) Omit the alt attribute entirely

05

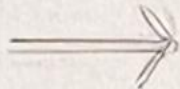
Objections “Best Of”

Really?

- “It slows us down.”
 - Accessibility debt ships slower. Ask Support.
- “Design will look worse.”
 - Design that fails outside the office isn’t design—it’s a mockup.
- “Screen reader users aren't our target demographic.”
 - Google Bot is the world's most active screen reader user.
- “We don’t know what to do!”
 - 90% of wins come from 10% of the rules: Semantic HTML and Contrast.

06

Future Mindset



2. **KAIZEN** (改善)

1% Better Every Day

Forget massive transformations.

Japanese companies dominate through tiny, daily improvements.

Today's micro-win becomes tomorrow's competitive advantage.




What's your 1% for today?

Thank you


get in touch ↘

Ariadne Engelbrecht

 Ariadne.Engelbrecht@inspired.consulting

 linkedin.com/in/ariadne-engelbrecht/



 [@ariadneengelbrecht](https://www.instagram.com/ariadneengelbrecht)

 curiosdevcookie.dev

  [@curiosdevcookie](https://www.github.com/ariadneengelbrecht)

List of sources of this talk

- WCAG by W3C: <https://www.w3.org/WAI/WCAG21/Understanding/>
- WCAG Testing Checklist: <https://www.accessguide.io/wcag-testing-checklist>
- Native Elements: <https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements>
- Good and bad a11y sites comparison: https://openassessittoolkit.github.io/accessible_u

List of great sites for accessibility

- The A11Y Project:
<https://www.a11yproject.com>
- A11y Cat:
<https://raindrop.io/a11ycat>
- A, AA, AAA:
<https://www.a11yproject.com/checklist/>
- All about screen readers: <https://tetralogical.com/blog/2025/10/14/common-misconceptions-about-screen-readers/>