# a11y… what?

## Web Accessibility

Julia Kirschenheuter

# Meine Motivation 🧗

- 🤼 Kooperation
- 🕸️ Vernetzung
- 🧘 Wissensaustausch

# WEB Accessibility

- 👥 Benutzbar für alle
- 💫 Unabhängig
- ⚠️ Alle öffentlichen Stellen der EU sind verpflichtet
- 🚀 Zeichen für hohe Qualität
- ⚠️ 2025: EU-Richtlinie 2019/882
- 🤕 Hand verletzt? Augen-OP? Keine Maus verfügbar? Brille weit weg?
- 📒 Kurze Geschichte

# Alles dreht sich um 🌎

- Inhalte sind:
  - wahrnehmbar (sichtbar)
  - bedienbar (nur bedienbare Interaktion)
  - verständlich (Informationen und Bedienung)
  - robust (interpretiert durch assistierende Techniken)

# Es bedeutet…

- Richtige Semantik
- Ohne Maus nutzbar (per Tastatur)
- Responsive (kein horizontales scrollen):
  - Inhalte brechen um ohne Verluste
  - Viewport 320px && 1280px + 400% zoom
- Ohne Bildschirm nutzbar (per Screenreader)
- + - Zooming
- Hohe Kontraste

# WCAG vs. BITV 🧐 🤔

| WCAG 2.1 \| 60 Prüfschritte | BITV 2.0 (EN 301 549) \| 98 Prüfschritte |
|---|---|
| https://ergebnis.bitvtest.de/pruefverfahren/wcag-21-web | https://ergebnis.bitvtest.de/pruefverfahren/bitv-20-web |
| https://www.w3.org/WAI/WCAG21/quickref/?currentsidebar=%23col_customize&versions=2.1 | |
| https://webaim.org/standards/wcag/checklist | |

- 👥 Basieren auf einem gemeinsamen Prüfverfahren
- 📍 BITV: +38 weitere Anforderungen
- ⚠️ Alle erfüllt oder nicht accessible
- ✅ Repräsentative Seitenauswahl
- 💸 Zertifizierung kostet viel Geld

# Nicht einfach

- Aufwand falsch verstanden
- Keine Planung && Aufwandsschätzung von Developern && Experten
- Zu wenig Fachkenntnisse
- Eine Wissenschaft an sich
- Sehr langwieriger Prozess
- Braucht viele Ressourcen
- Jeder Versuch ist teuer
- Viel 🤯 🩸 😰

# Was tun?

Extra Version für sehschwächere Menschen

Plugin nehmen

Fertige lib nehmen?

Eigene Lib entwickeln

Wrapper bauen

# Herangehensweise (gilt immer)

- 🧠 Komplexität der Logik und Umsetzung
- 👨‍💻 Recherche der Anforderungen
- 📍 Planung + Tests
- ⚒️ Nur UI Komponente

# Version für sehschwächere Menschen

- Löst nur 1 Teil des Problems
- Meistens optisch nicht ansprechend
- Getrennte Version (doppelte Pflege)
- https://stadtwerke.herrenberg.de/gaubader/

# Accessibe & co. 💡

- Nicht accessible 😂😂
- Schreibt den Code nicht um
- https://accessibe.com/

# Neu entwickeln / Wrapper bauen

Teils von Null an schreiben / teils „wrappern“

https://nextcloud-vue-components.netlify.app/

# „to go“ lib? 🧐🤔

- Mit Vorsicht genießen, doppelt und dreifach prüfen!
- https://carbondesignsystem.com/
- https://reach.tech/
- https://reakit.io/
- https://codyhouse.co/

# Noch besser...

Gleich an a11y denken

Im System integrieren

# a11y vs Design

# Besonders hart 🥵

- Toolitps, https://github.com/nextcloud/nextcloud-vue/issues/2503
- DatetimePicker, https://github.com/nextcloud/nextcloud-vue/issues/1666
- Lazy loading, https://github.com/nextcloud/server/issues/33844
- Bestehende Ansichten / Komponenten umschreiben

# Ein paar Fallen

- Ebay & Amazon
- Off screen focus (sidebar nav tabs)
- Focus trap
- Page structure (headings, landmarks)

# Headings

- Headings richtig integrieren
- Semantik | Aussehen trennen (As="h2" Size = „s")
- Noch nicht accessible:

https://nextcloud-vue-components.netlify.app/#/Components/NcEmptyContent

# ⚒ Werkzeuge zum testen 🛠

- https://github.com/JuliaKirschenheuter/accessibility/

# Automatisieren 🚀

- Unit Tests
- Integration Tests
- jest-axe
- jest-dom
- axe-core
- testing-library

# Mitdenken (nice to have)

- Leichte Sprache
- Gebärdensprache
- Leichte && verständliche Struktur

# Sonstige Vorteile

- 😸 Qualitative Web Umsetzung
- 🐞 Weniger Bugs
- 💫 CEO, Rating
- 📈 Bessere Performance
- ✍️ Gesetzliche Vorgaben erfüllt
- 👥 Sozial Verantwortlich
- ✅ Zertifizierung

# Bereits accessible 💡💡

- https://www.bitvtest.de/sites_und_agenturen/barrierefreie_websites.html

# Ihr findet mich unter

- https://github.com/JuliaKirschenheuter/
- https://www.linkedin.com/in/julia-kirschenheuter-a61459200/