



Opentalk

Barrierefreiheit bei OpenTalk



Konkrete Umsetzung von Barrierefreiheit anhand von
Beispielen

Wir haben Zeit!

Wir tauschen uns aus.

Fragen? Jederzeit bitte!



Wer bin ich?



Emre Neumann

Geboren in Wien

Wohne in Friesland

Frontend-Entwickler seit 8 Jahren

Bei OpenTalk seit 2024

A11y war 2024 das dominante Thema bei OpenTalk

Barrierefreiheit



How it started

PRÜFBERICHT
OpenTalk
Barrierefreiheit

9.4.1.2 Name, Rolle, Wert verfügbar (Meeting-Details Button)

Mangel	Die Auszeichnung des Info-Buttons beim Meeting-Titel g... Button ein Modal-Fenster öffnet. Zusätzlich ist der Altern...
Empfehlung	Es sollte eine Auszeichnung mit aria-haspopup="dialog"- Zusätzlich wird empfohlen den Alternativtext anzupasser missverstanden werden kann. Passender wäre „Meeting „Meetingdetails anzeigen“

How it's going

Open 16 Closed 193 All 209

🕒 Label is Accessibility X



Marco
@marco@chaos.social

Habe soeben eine halbe Stunde lang mit @andijah und @SandiPavkovic die Funktionen von @OpenTalkMeeting mit Screen Readern und sehender Hilfe getestet. Wir konnten tatsächlich alles erreichen und auslesen, inkl. Chats, einer Umfrage, Teilnehmerliste usw. Nur Protokolle und Whiteboards scheinen bei der Installation bei @mailbox_org nicht aktiviert zu sein, beide Funktionen waren nicht auffindbar. Ton und Video liefen super.

Was haben wir gelernt?



- Screenreader
- Tastatursteuerung
- Farben und Kontraste
- Zoom
- Tooltips & Icons
- Tools die wir entdeckt haben



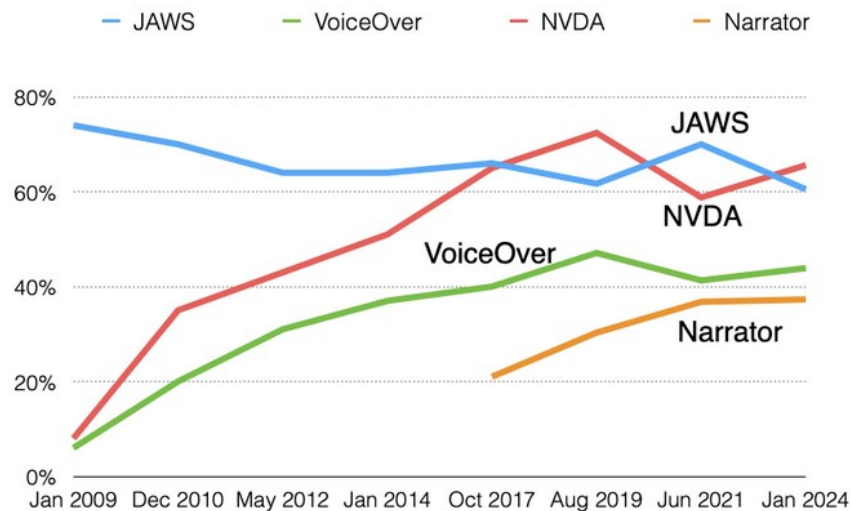
Screenreader

- Was ist ein Screenreader?
- Welche Screenreader gibt es?
- Wie mach ich mein HTML maschinenlesbar?
- Was haben gute Tests mit Screenreadern zu tun?
- Beispiele

Welche Screenreader gibt es?



- Windows: JAWS*, NVDA, Narrator
- Mac: VoiceOver
- Linux: Orca



Which of the following desktop/laptop screen readers do you commonly use?

Response	# of respondents	% of respondents
NVDA	1009	65.6%
JAWS	931	60.5%
VoiceOver	675	43.9%
Narrator	574	37.3%
Orca	127	8.3%
ZoomText/Fusion	115	7.5%
Dolphin SuperNova	83	5.4%
ChromeVox	59	3.8%
System Access or System Access to Go	28	1.8%
Window-Eyes	18	1.2%
Other	79	5.1%

Quelle: <https://webaim.org/projects/screenreadersurvey10/>

*kostenpflichtig

Screenreader



- QA testet hauptsächlich mit NVDA
- Devs testen mit Orca oder VoiceOver
- Screenreader verhalten sich leider unterschiedlich
- Screenreader sind konfigurierbar
- Fazit: semantisch korrektes HTML bauen und den A11Y-Tree regelmäßig checken

Maschinenlesbarkeit

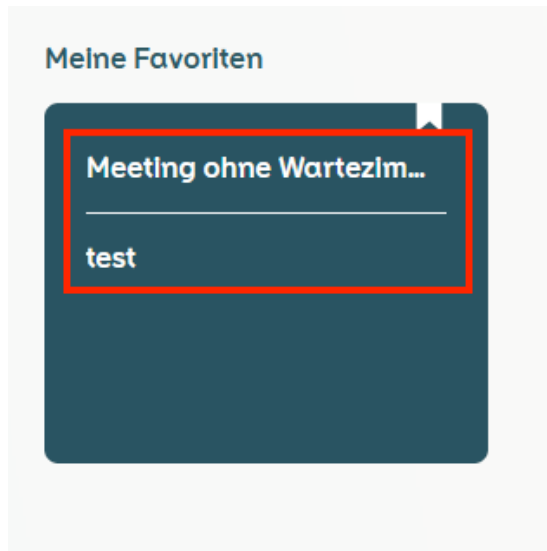


Semantisch korrektes HTML benutzen

- HTML Spezifikationen umsetzen
- z.B. keine `<div>` mit clickHandler
- Header Reihenfolgen einhalten (h1 > h2 > h3)
- Listen sollten `` oder `` verwenden
- Tabellen sollten Daten anzeigen
- `<nav>` `<main>` `<aside>` `<dialog>` `<menu>` ...



Beispiel



Vorher

```
<p> Meine Favoriten </p>
<a> Meeting ohne Wartezim ... </a>
<a> test </a>
```

Nachher

```
<h3> Meine Favoriten </h3>
<ul>
  <li><a> Meeting ohne Wartezim ... </a></li>
  <li><a> test </a></li>
</ul>
```

Tests und Screenreader



- Das Verhalten von Screenreadern ist schwer vorhersehbar
- E2E oder Unit Tests die sich an vorgegebene Richtlinien halten geben Sicherheit

Tests und Screenreader



Beispiel: Schlechter Test

```
21 describe('OverviewCard', () => {
22   it('renders without crashing', () => {
23     renderWithProviders(<OverviewCard isMeetingCreator={false} event={mockedMeeting} />, {
24       provider: { router: true, mui: true },
25     });
26
27     expect(screen.getByTestId('MeetingOverviewCard')).toBeInTheDocument();
28     expect(screen.getByRole('img', { name: 'global-favorite' })).toBeInTheDocument();
29   });
}
```

Tests und Screenreader



Besser

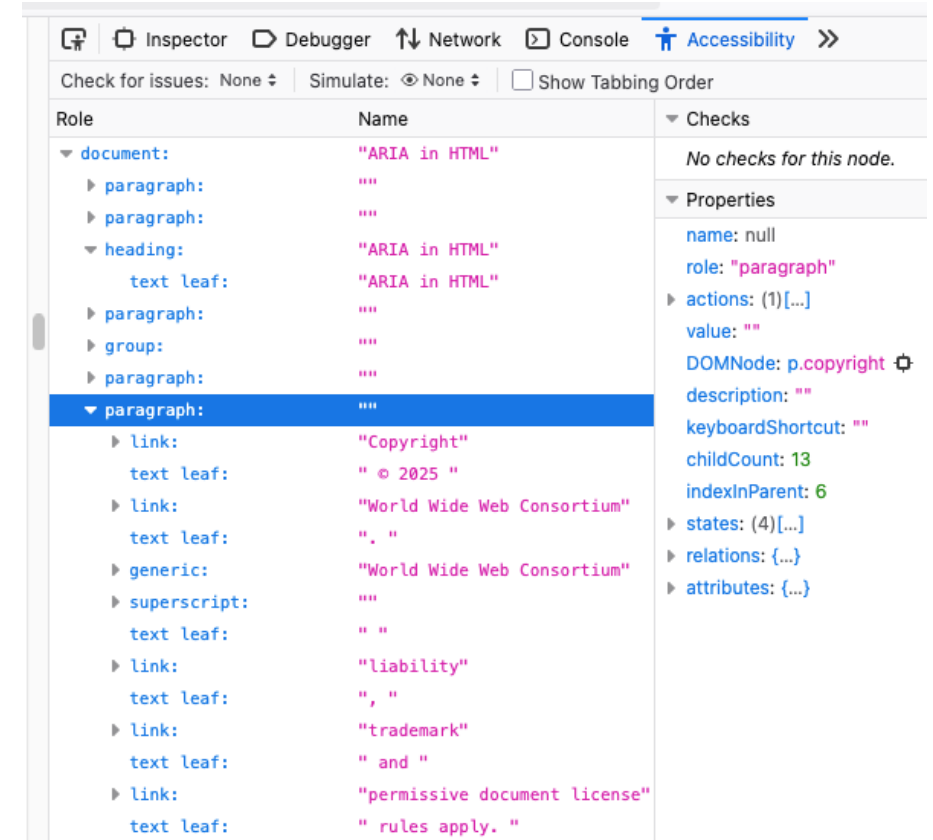
```
21 describe('OverviewCard', () => {
22   it('renders without crashing', () => {
23     renderWithProviders(<OverviewCard isMeetingCreator={false} event={mockedMeeting} />, {
24       provider: { router: true, mui: true },
25     });
26     const overviewCard = screen.getByRole('heading', { name: eventMockedData.title });
27     expect(overviewCard).toBeInTheDocument();
28
29     expect(screen.getByRole('img', { name: 'global-favorite' })).toBeInTheDocument();
30   });
31 }
```

Tests und Screenreader: Fazit



Wer `getByRole` verwendet ist auf der sicheren Seite

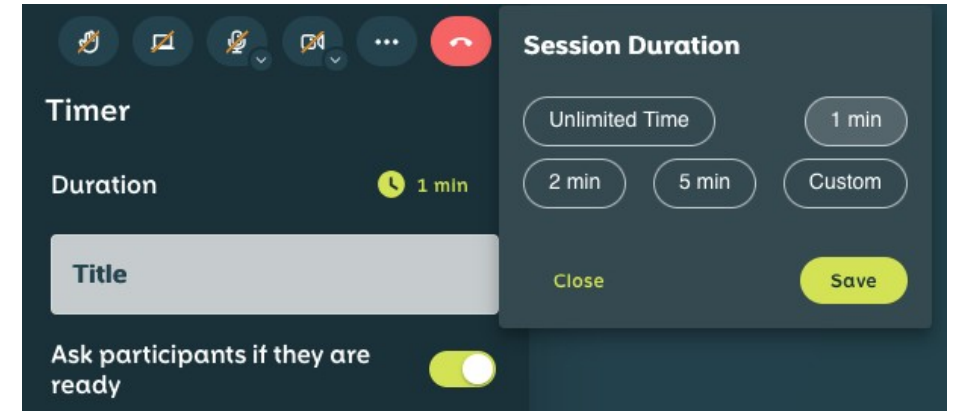
`getByRole` lokalisiert Elemente im Accessibility Tree genau wie es ein Screenreader tut



Beispiel: Timer einstellen



Problem: Button-label kürzt
das Wort „Minute“ ab
Screenreader liest „m“-“i“-“n“





Beispiel: Timer einstellen

Lösung: „Minute“ in aria-label ausschreiben

→ `<button>`

```
<button tabindex="0" aria-  
selected="true" aria-label="1 minute">  
</button>
```

Tastatursteuerung



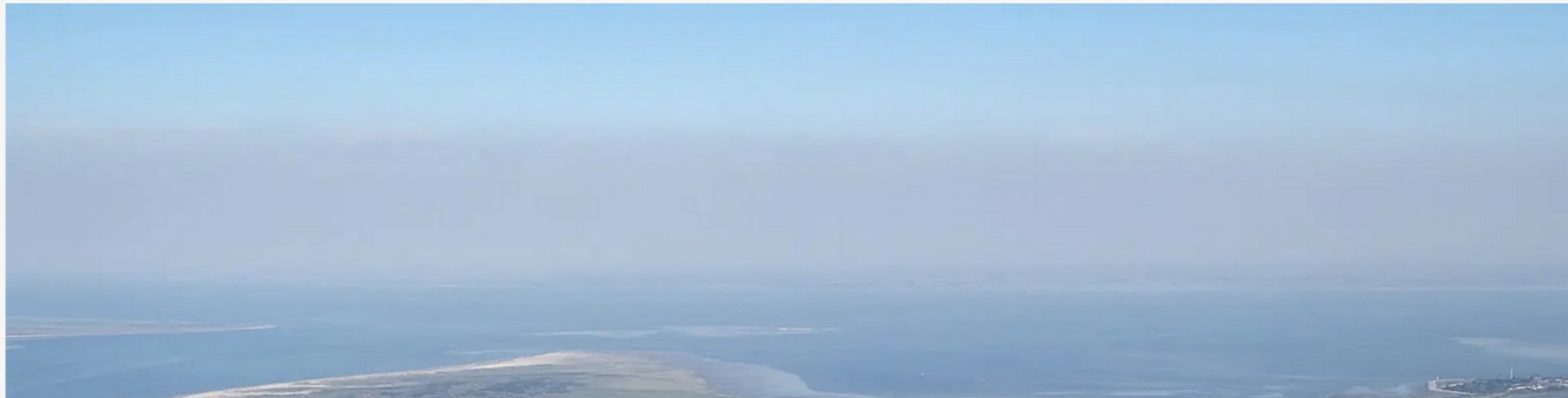
- Was sind Skip Links?
- Fokus
- Das Space und Enter Problem

Skip Links



Hauptnavigation NDR

[Nachrichten](#) [Niedersachsen](#) [Schleswig-Holstein](#) [Mecklenburg-Vorpommern](#) [Hamburg](#) [Sport](#) [Ratgeber](#) [Kultur](#)



Skip Links



- Auch jump links genannt
- Sprung zu bestimmten Bereichen auf der Seite



Fokus



- Styling für fokussierte Elemente
- Tabreihenfolge
- Auto Fokus

Das Space und Enter Problem



- Space ist bei uns Push-to-Talk
- Enter löst nicht jedes Element aus
- Tastaturkürzel bei OpenTalk lassen sich deaktivieren

Farben und Kontraste



- Farbkontrast
- OKLCH

Farbkontrast



- Wichtig für Lesbarkeit von Text

The screenshot displays a 'Colour contrast checker' interface. At the top, there is a message input field with the placeholder 'type a message...' and a red error message: 'Error: Empty messages are not allowed'. Below this, the tool title 'Colour contrast checker' is shown in orange. A sample text 'Aa' is displayed in a large font, with a contrast ratio of '3.55' next to it. Below the text, there are four status indicators: 'Pass' with a checkmark, and three 'Fail' indicators with 'X' marks, corresponding to 'AA Large', 'AAA Large', 'AA Normal', and 'AAA Normal'. A 'Save colours' button is located below the status indicators. To the right, there are two color selection panels: 'Background colour' with the hex code '#17303a' and 'Foreground colour' with the hex code '#ee3f33'. Each panel includes RGB and HSL sliders. The RGB values for the background are Red 23, Green 48, and Blue 58. The RGB values for the foreground are Red 238, Green 63, and Blue 51. A 'Buy me a coffee' button with a coffee cup icon is also visible in the top right corner of the tool interface.

OKLCH Farbraum



- Ähnlich HSL
- Wird von CSS unterstützt
- L in OKLCH entspricht der wahrgenommenen Helligkeit (anders als bei HSL)
- Ermöglicht das berechnen von Farbtönen

Zoom



Zielgruppen:

- Personen mit (stark) eingeschränkter Sicht
- User mit anderen Endgeräten

Zoom



Das Layout muss bei 200% 300% Zoomstufe auch noch funktionieren

Tooltips & Icons



Tooltips & Icons



Tooltips:

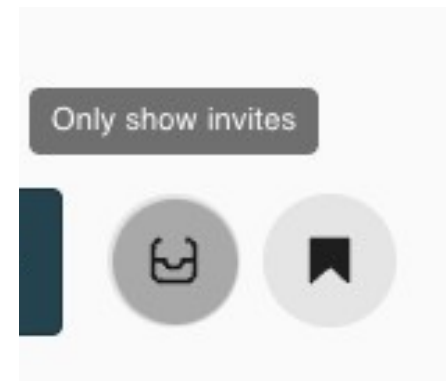
- Exklusiv für Maususer
- Information im Tooltip muss Tastaturusern zugänglich sein
- Wir haben viele Tooltips entfernt

Tooltips & Icons



Icons:

- Buttons mit ausschließlich Icon-Label brauchen aria-label attribut (und Tooltips ;))



Tools



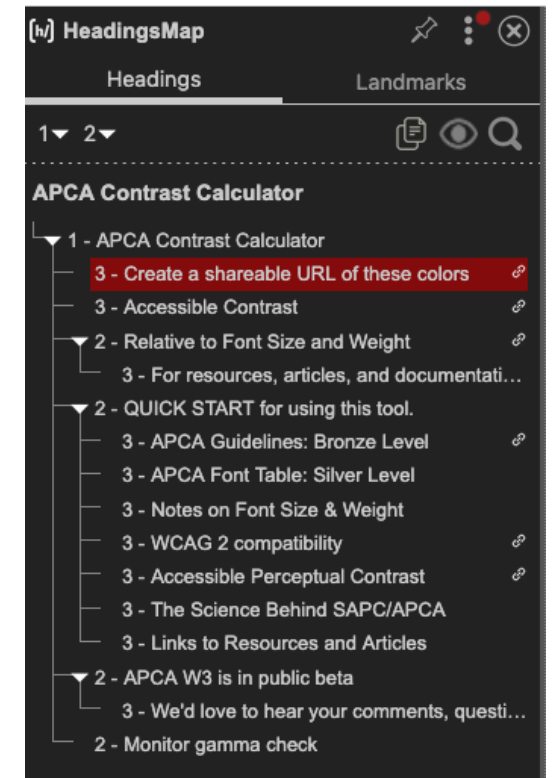
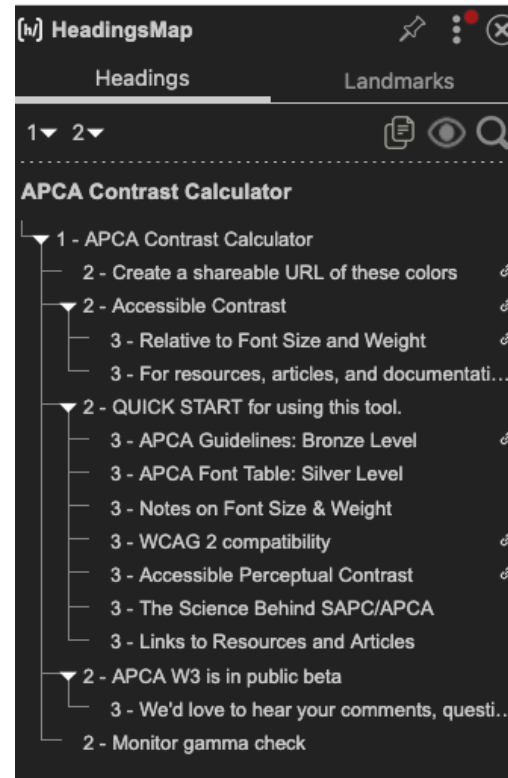
- [Headings Map](#)
- [Colour Contrast Checker Plus](#)
- [APCA Contrast Calculator](#)

Headings Map



Browser Extension

- Sucht alle <h> tags raus
- Zeigt Fehler in der Hierarchie an



Colour Contrast Checker Plus



Browser Addon zum Überprüfen des Farbkontrasts

type a message...
Error: Empty messages are not allowed

Colour contrast checker

Buy me a coffee

Aa 3.55

Pass ✓ Fail ✗ Fail ✗ Fail ✗
AA Large AAA Large AA Normal AAA Normal

Save colours Aa Aa Aa

Background colour: #17303a

Foreground colour: #ee3f33

Background colour (RGB): Red 23, Green 48, Blue 58

Foreground colour (RGB): Red 238, Green 63, Blue 51

APCA Contrast Calculator



[Demo ...](#)



OpenTalk
Fragen, Wünsche
Anregungen