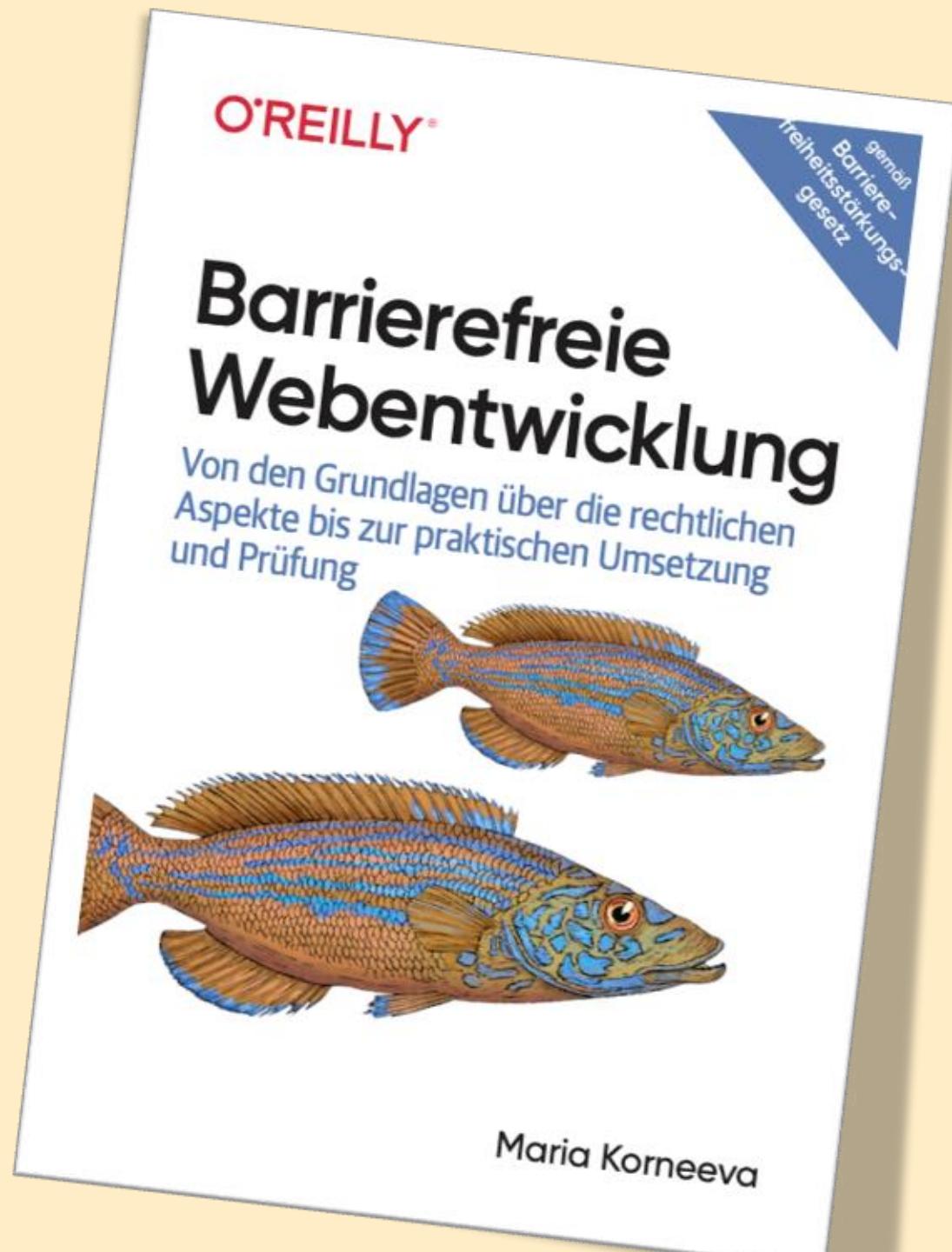


INTERAKTIVE ELEMENTE, DIE SPRECHEN

Maria Korneeva @ Plätzchen-Talk



MARIA KORNEEVA

Frontend Technology Lead

Google Developer Expert

Trainer since 2022

Book author since 2024



/maria-korneeva



kontakt@barrierefreie-
webentwicklung.com

BUTTON



Hallo!

DIESE INTERAKTIVEN ELEMENTEN KÖNNEN SELBST SPRECHEN:

absenden

HIER

speichern

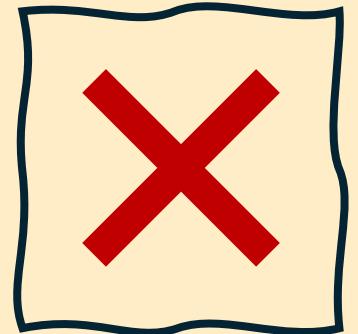
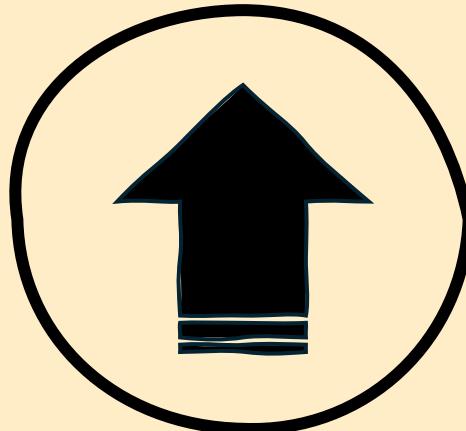
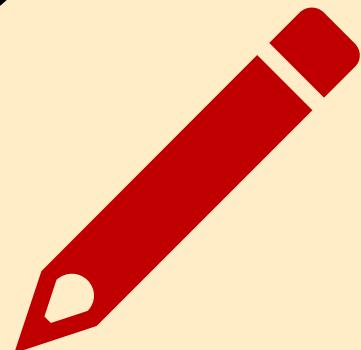
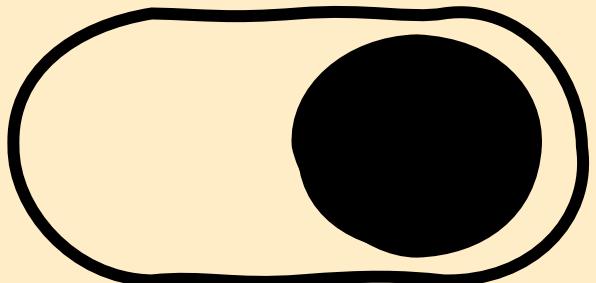
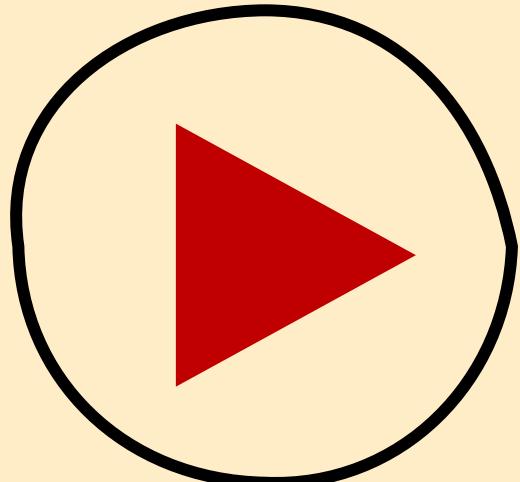
ZAHLUNGSPFLECHTIG BEANTRAGEN

ZURÜCK

JETZT BUCHEN !

weiter >

DIESE INTERAKTIVEN ELEMENTE KÖNNEN NICHT SELBST SPRECHEN:



...

WAS IST DER UNTERSCHIED?

BEARBEITEN

```
<button>  
  Bearbeiten  
</button>
```

| | | | |



```
<button>  
  <img src=.../>  
</button>
```

DER ZUGÄNGLICHE NAME

Die **Bezeichnung eines Elements**, die von assistiven Technologien (z.B. **Screenreadern**) angekündigt wird. Damit ist klar, **was das Element ist oder wofür es dient**.

WAS IST DER UNTERSCHIED (ZUGÄNGLICHER NAME) ?

BEARBEITEN

Mit zugänglichem Namen:
“button, bearbeiten”

| | | | |

Ohne zugänglichem Namen:
“button”



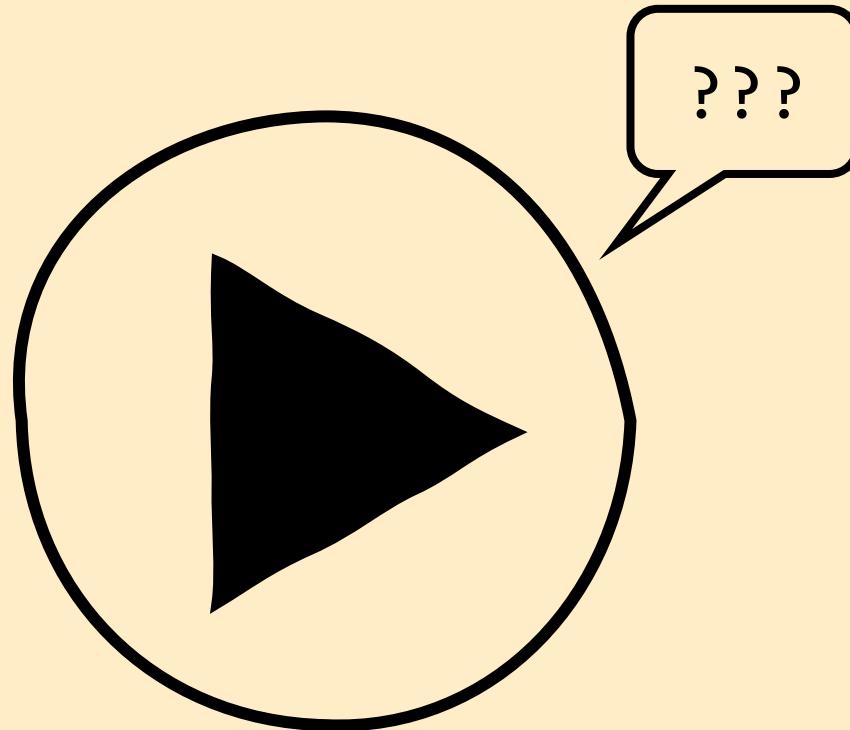
WARUM IST DER ZUGÄNGLICHE NAME WICHTIG?

- Blinde, sehbehinderte Menschen wissen sonst nicht, was das interaktive Element tut
- Sprachsteuerung funktioniert sonst nicht (zuverlässig)
- Potenzieller Verstoß gegen Web Content Accessibility Guidelines Erfolgskriterium 4.1.2 „Name, Rolle, Wert verfügbar“ (BFSG, BITV 2.0)

WIR MÜSSEN ALSO AGIEREN – IM DESIGN

- Im **Design**-Prozess gleich den zugänglichen Namen bedenken (ggf. Figma-Annotationen nutzen)

BRAUCHE ICH WIRKLICH ICON-BUTTONS?



WIR MÜSSEN ALSO AGIEREN – IN DER FACHVORGABE

- Im **Design**-Prozess gleich den zugänglichen Namen bedenken (ggf. Figma-Annotationen nutzen)
- Als **Fachexperte** den zugänglichen Namen festlegen und in der Fachvorgabe vermerken

WAS IST EIN GUTER ZUGÄNGLICHER NAME (1) ?

- **Kurz** und **prägnant** (“Sie erfahren in unserem Leitfaden, wie Sie Fördermittel beantragen.” statt “Wie Sie Fördermittel beantragen, erfahren Sie hier.”)
- Zugängliche **Rollen** (z.B. “Button”, “Bild von...”, “Link”) sind im zugänglichen Namen nicht nötig
- Das wichtigste **zuerst** (“Zum Formular (neuer Tab)” statt “(Neuer Tab) Zum Formular”)
- Beschreiben Sie die **Funktion** bzw. das Ziel statt des visuellen Inhalts (“Suchen” statt “Lupe”)

WAS IST EIN GUTER ZUGÄNGLICHER NAME (2) ?

- Achten Sie auf die **einheitliche Bezeichnung** (z.B. überall “Entfernen” statt “Löschen/Entfernen”)
- Sorgen Sie für **eindeutige Namen** bei gleichen Buttons/Links (“Vertrag ABC bearbeiten”, “Vertrag XYZ bearbeiten” etc.)
- Starten Sie mit einem **Großbuchstaben**, damit die Screenreader den Namen mit richtiger Intonation aussprechen. Setzen Sie **keinen Punkt** am Ende.
- Vergessen Sie nicht, die zugänglichen Namen zu **übersetzen**

WIR MÜSSEN ALSO AGIEREN – IM CODE

- Im **Design**-Prozess gleich den zugänglichen Namen bedenken (ggf. Figma-Annotationen nutzen)
- Als **Fachexperte** den zugänglichen Namen festlegen und in der Fachvorgabe vermerken
- Den zugänglichen Namen **im Code** festhalten

ZUGÄNGLICHER NAME IM CODE (1)

```
<button>  
  <img alt="Bearbeiten"/>  
</button>
```

ZUGÄNGLICHER NAME IM CODE (2)

```
<button>
  <span class="sr-only">Bearbeiten</span>
  ...
</button>
```

```
.sr-only {
  width: 1px; height: 1px;
  overflow: hidden; clip-path: inset(50%);
  position: absolute; white-space: nowrap;
}
```

ZUGÄNGLICHER NAME IM CODE (3)

```
<p id="p-name">Bearbeiten</p>  
<button  
aria-labelledby="p-name" >  
...  
</button>
```

ZUGÄNGLICHER NAME IM CODE (4)

```
<button  
  aria-label="Bearbeiten" >  
  ...  
</button>
```

MEHRERE WEGE FÜHREN NACH ROM

```
<p id="p-name">Name 1</p>
<button aria-labelledby="p-name"
aria-label="Name 2" title="Name 3" >
Name 4
</button>
```

- Spezifikation: Accessible name and description computation 1.2

ÜBERSCHREIBEN SIE NICHT DEN SICHTBAREN TEXT

```
<button  
  aria-label="Kann rückgängig gemacht werden">  
  Bearbeiten  
</button>
```

Kann rückgängig...

BEARBEITEN

WIE IMPLEMENTIERE ICH EINEN ZUGÄNGLICHEN NAMEN?

- Nutzen Sie möglichst **native Methoden** (alt-Attribut oder visuell versteckte Inhalte)
- **Überschreiben** Sie bereits vorhandenen sichtbaren Text **nicht mit ARIA-Technicken** (aria-label, aria-labelledby)
- **Testen** Sie im Browser und mit Screenreader

BEISPIEL

UND NUN?

NÄCHSTE SCHRITTE

- Ersetzen Sie Icon-Buttons/Links durch interaktive Elemente mit sichtbarem Text, wenn möglich.
- Identifizieren Sie ein interaktives Element, bei dem der zugängliche Name fehlt oder überschrieben wird und adressieren Sie das Problem im Team
- Icon-Buttons und -Links werden Ihnen „danke“ sagen!



FRAGEN?

KONTAKT@BARRIEREFREIE-WEBENTWICKLUNG.COM