SOP 24

Navigationstexte & Linkbezeichnungen

Projektleitung | Geschäftsführung | IT | Recht

Was du hier machst	Du sorgst dafür, dass alle Links und Navigationselemente auf deiner Website auch ohne visuellen Kontext verständlich und zugänglich sind – für alle Nutzer:innen, besonders für Screenreader-Anwender:innen.
Voraussetzung	Deine Website hat klickbare Elemente (z. B. Navigation, Footer, Teaser etc.) und du kennst die allgemeinen Anforderungen an barrierefreie Inhalte (vgl. SOP 1).
Dauer / Aufwand	Je nach Umfang der Website 2–8 Stunden (inkl. Prüfung & Korrektur) 🩹 Pflicht? Ja – laut EAA/BFSG und WCAG 2.1 verpflichtend für alle betroffenen Websites ab dem 28. Juni 2025
Gesetzlich verpflichtend	✅ Ja – laut EAA/BFSG und WCAG 2.1 verpflichtend für alle betroffenen Websites ab dem 28. Juni 2025
Nächster Schritt	→ SOP 25: Formulare und Buttons

🗒️ SOP 24: Navigationstexte & Linkbezeichnungen barrierefrei gestalten

So werden aus vagen Klicks klare Wegweiser

Wer sollte diese SOP lesen?

Was du hier machst

Du sorgst dafür, dass alle Links und Navigationselemente auf deiner Website auch ohne visuellen Kontext verständlich und zugänglich sind – für alle Nutzer:innen, besonders für Screenreader-Anwender:innen.

🧠 Voraussetzung

Deine Website hat klickbare Elemente (z. B. Navigation, Footer, Teaser etc.) und du kennst die allgemeinen Anforderungen an barrierefreie Inhalte (vgl. SOP 1).

Ergebnis

Eine Website, deren Linktexte klar, kontextunabhängig verständlich und technisch korrekt ausgezeichnet sind – geprüft und ready für die Barrierefreiheitsprüfung.

⏱️ Dauer

Je nach Umfang der Website 2–8 Stunden (inkl. Prüfung & Korrektur)

🩹 Gesetzlich verpflichtend?

✅ Ja – laut EAA/BFSG und WCAG 2.1 verpflichtend für alle betroffenen Websites ab dem 28. Juni 2025

📍 Nächster Schritt

→ SOP 25: Formulare und Buttons

Warum machst du das überhaupt?

Die Navigation (Buttons, Links, etc.) ist das Rückgrat jeder Website, aber nur, wenn man sie verstehen kann.

Und wenn du Leute mit „Hier klicken“ und ähnlichem zum navigieren aufforderst, hilft das niemandem – auch nicht deinen BesucherInnen ohne Einschränkungen.

Für deine BesucherInnen MIT Einschränkungen allerdings machst du die Navigation auf deiner Seite nicht nur problematisch, sondern nahezu unmöglich. Denn für Screenreader-Nutzer:innen, die sich eine Liste aller Links auf einer Seite vorlesen lassen können, ist ein Link nicht etwas, das sie sehen, sondern hören müssen.

Ohne eine klare Bezeichnung, die das Ziel beschreibt, hören sie aber nur einen Dschungel aus „Mehr“, „Weiter“ oder „Hier“.

Barrierefreiheit bedeutet hier also: Links müssen sagen, wohin sie führen, nicht nur, dass man klicken soll.

Am Ende hast du einen klaren Prozess, um sicherzustellen, dass alle klickbaren Texte auf deiner Website und in deiner Navigation eindeutig und kontextunabhängig verständlich sind. Nutzer:innen können sich so effizient orientieren und wissen immer, wohin ein Link sie führt.

Du legst deine Farbpalette in demselben Ordner wie alle deine Dokumentationen und Richtlinien zur Barrierefreiheit ab, deinem Barrierefreiheitsordner.

- Je nach Umfang der Website 2–8 Stunden (inkl. Prüfung & Korrektur)
- Content: Benennt die Navigationselemente und formuliert alle Linktexte klar und eindeutig.
- Dev / Design: Setzt die Links technisch und visuell korrekt um.
- UX / QA: Prüft die umgesetzten Links auf Verständlichkeit und Barrierefreiheit.

Wann musst du das machen?

Bei allen klickbaren Texten auf deiner Website, besonders in:

- Hauptnavigation & Footer
- Linklisten und Teaserboxen
- Inhaltsverzeichnissen
- Blogvorschauen und Produktkacheln
- PDF-Download-Links
- Call-to-Action-Elementen in Formularen

Welches Gesetz verlangt das?

- European Accessibility Act (EAA) / Barrierefreiheitsstärkungsgesetz (BFSG): Fordern, dass Nutzer:innen sich ohne Sehen orientieren können müssen.
- WCAG 2.1: Insbesondere die Erfolgskriterien:
- 2.4.4 Link Purpose (In Context)
- 2.4.9 Link Purpose (Link Only)

Der Prozess im Detail

1 Formuliere klare und aussagekräftige Linktexte

Als Erstes nimmst du dir deine Link-Texte vor.

Die Grundregel lautet: Ein Linktext muss auch dann noch verständlich sein, wenn man ihn aus dem Kontext reißt. Vermeide darumunbedingt generische Bezeichnungen.

- Ein guter Linktext ist aufrichtig, substanziell, prägnant und spezifisch.
- Schlecht: „Hier klicken“ → Gut: „Produktdetails anzeigen“
- Schlecht: „Mehr“ → Gut: „Mehr über unsere Trainings erfahren“
- Schlecht: „Download“ → Gut: „PDF mit Preisübersicht herunterladen (2 MB)“
- Schlecht: „Weiterlesen“ → Gut: „Weiterlesen: Barrierefreiheit erklärt“

2 Benenne die Navigation eindeutig

Auch die festen Navigationselemente müssen klar sein. Verwende niemals nur Icons ohne begleitenden Text für die Navigation. Die Grundegel hier lautet: Je spezifischer du bist, desto besser ist es für die Barrierefreiheit.

- Hauptnavigation: Nutze „Über uns“ statt nur „Info“.
- Footer-Links: Schreibe „Kontaktformular“ statt nur „Kontakt“.
- Breadcrumbs: Formuliere den Pfad vollständig aus, z. B. „Startseite > Produkte > Produkt 1“.
- Wann immer möglich, sollte die Hauptnavigation an einer vorhersehbaren Stelle auf der Seite platziert werden, z.B. am oberen oder linken Rand (für linksläufige Sprachen).

3 Stelle sicher, dass die Links vorgelesen werden können

Nachdem du den sprachlichen Teil abgearbeitet hast, musst du nun sicherstellen, dass deine Links im Hintergrund auch technisch korrekt funktionieren und für alle zugänglich sind. Das ist wichtig, damit deine Links auch von Hilfsmitteln wie Screenreadern oder Tastaturen verstanden werden.

- Verwende immer die richtige „Link-Grundlage“: <a href=“…“> Das ist das Standardelement für Links im Web. Wenn du z.B. einen Link zu dieser SOP setzen willst, verlinkst du einen Text. In deinem Website Code steht am Ende <a href=“https://regukit.com/sop-24″> Browser und Hilfstechnologien wie Screenreader erkennen dieses Element automatisch als Link. Sie wissen dann, dass man darauf klicken kann und dass es zu etwas Neuem führt. Wenn du stattdessen oder verwendest und diese nur mit einem Trick klickbar machst, verstehen Screenreader das oft nicht, und Tastaturnutzer können sie möglicherweise nicht bedienen. Dein Link wäre dann wie eine unsichtbare oder klemmende Tür. Guck dir also deinen Quelltext an und stelle sicher, dass dein Link mit <a href=“…“> gekennzeichnet ist.
- Füge deinem Link dann ein Aria-Label hinzu. Dazu nimmst du deinen Link und fügst ein Aria-Label und eine Beschreibung hinzu. Unser Link sieht dann wie folgt aus: Regukit SOP 24. Wenn nun jemand auf deiner Seite ist (auf der sich dieser Link befindet, sehen reguläre Besucher einfach nur einen Link mit dem Text „Regukit SOP 24“. Sehbehinderte Menschen allerdings, die einen Screenreader nutzen, bekommen „Hier geht es zur SOP Nummer 24“ vom Screenreader vorgelesen.
- Wichtig für dich: Sei sehr vorsichtig mit aria-label! Es überschreibt den normalen Linktext für Screenreader. Wenn du unsicher bist, wie du diese Attribute korrekt einsetzt, sprich bitte mit deinem Entwickler.

4 Stelle sicher, dass die Links per Tastatur gesteuert werden können

Natürlich dürfen wir nicht nur an Menschen denken, deren Sehvermögen eingeschränkt ist, sondern auch an jene, die keine Maus bedienen können. Für sie ist es entscheidend, dass alle Links auch per Tastatur erreichbar sind – zum Beispiel über die Tab-Taste.

Bei einem klassischen Link ist das kein Problem: HTML-Links mit einem gültigen href sind automatisch fokussierbar und funktionieren wie erwartet – auch mit Tastatur und Screenreader.

Aber: Es gibt Fälle, in denen ein Link ohne href -Attribut eingebunden wird – etwa so:

< a onclick = "doSomething()" >Mehr erfahren </ a >

In solchen Fällen erkennt der Browser das Element nicht als echten Link .

Das bedeutet: Kein Fokus per Tab , keine Tastaturbedienung , keine semantische Erkennung durch Screenreader .

Die beste Lösung :

Solche Elemente nicht als „Fake-Link“ einbauen , sondern direkt als echtes HTML-Element , z. B. ein <button> oder ein <a href="#"> mit klarer ARIA-Rolle und tabindex .

Denn: Es gibt keinen guten Grund , einen Link ohne href zu setzen – und es ist technisch immer möglich, eine barrierefreie Lösung zu finden, ohne auf semantisch unsaubere Konstruktionen zurückzugreifen .

Wenn du einen Link ohne href verwendest, ist es kein echter Link – weder für Browser, noch für Screenreader oder Tastaturnutzer. Vermeide das. Verwende stattdessen echte HTML-Elemente mit klarer Funktion.

5 Denke auch an Sonderfälle wie Linklisten und Kachel-Teaser

Auch bei visuellen Elementen, die als Ganzes klickbar sind, muss das Ziel für Screenreader klar sein.

- Blog-Teaser: Verlinke nicht nur ein vages „Mehr erfahren“, sondern den gesamten Titel oder formuliere den Linktext als „Mehr erfahren: [Titel des Artikels]“. Damit erleichterst du den Zugang zu weiteren Informationen, indem du eine große, informative Klickfläche anbietest.
- Produktkachel: Wenn eine Kachel mit Bild, Preis und Button klickbar ist, gib dem CTA-Button eine sinnvolle Bezeichnung, z. B. „Jetzt kaufen: Produkt 1“.
- Pagebuilder-Vorsicht: Achte in Tools wie Divi oder Elementor darauf, dass nicht die gesamte Box ohne klaren Fokus verlinkt wird, sondern das konkrete Call-to-Action-Element. So lenkst du den Fokus auf die spezifische Aktion und vermeidest unnötige, unklare Klickflächen. Der Nutzer soll klar erkennen, wo die Interaktion stattfinden soll.

Am Ende dieser SOP hast du:

- Linktexte, die auch ohne visuellen Kontext klar verständlich sind.
- Eine Navigation, die für alle Nutzer:innen leicht erfassbar und bedienbar ist.
- Technisch korrekt ausgezeichnete Links, die die Bedienung mit Hilfsmitteln sicherstellen.
- Ein geringeres Risiko für EAA-Verstöße und eine bessere User Experience für alle.

Copy-Paste-Baustein

Infos zur Barrierefreiheit von Bildern & Grafiken

Du hast deine Links überarbeitet? Super! Mithilfe dieser Checkliste kannst du jetzt schnell überprüfen, ob du an alles gedacht hast und deine Links wirklich für jeden zugänglich sind.

Schritt 1: Klare und aussagekräftige Linktexte

Stelle sicher, dass deine Linktexte verständlich sind, auch wenn man sie aus dem Kontext reißt.

- [ ] Ist der Linktext aufrichtig und substanziell? (Z.B. „Produktdetails anzeigen“ statt „Hier klicken“)
- [ ] Ist der Linktext prägnant und spezifisch? (Z.B. „PDF mit Preisübersicht herunterladen (2 MB)“ statt „Download“)
- [ ] Vermeidest du generische Bezeichnungen? (Z.B. „Mehr über unsere Trainings erfahren“ statt „Mehr“)

Schritt 2: Eindeutige Navigation

Sorge dafür, dass deine Navigationselemente klar und verständlich sind.

- [ ] Verwendest du immer Text zu Icons in der Navigation? (Nie nur Icons ohne Text!)
- [ ] Sind deine Hauptnavigationselemente spezifisch benannt? (Z.B. „Über uns“ statt „Info“)
- [ ] Sind deine Footer-Links eindeutig? (Z.B. „Kontaktformular“ statt „Kontakt“)
- [ ] Sind deine Breadcrumbs vollständig ausformuliert? (Z.B. „Startseite > Produkte > Produkt 1“)
- [ ] Ist die Hauptnavigation an einer vorhersehbaren Stelle platziert? (Oben oder links)

Schritt 3: Links können vorgelesen werden

Überprüfe, ob deine Links technisch korrekt funktionieren und von Hilfsmitteln verstanden werden.

- [ ] Verwendest du immer das $\\texttt{\ }$ Element für Links? (Kein $\\texttt{\ oder $\\texttt{\ !)
- [ ] Wurde t e x ttt a r ia − l ab e l mit Bedacht und nur bei Bedarf hinzugefügt? (Denk daran: Es überschreibt den normalen Linktext für Screenreader!)
- [ ] Ist der t e x ttt a r ia − l ab e l Text kurz, prägnant und beschreibt das Linkziel klar?

Schritt 4: Links können per Tastatur gesteuert werden

Sicher, dass alle Links auch ohne Maus bedienbar sind?

- [ ] Besitzt jeder Link ein gültiges t e x ttt h re f -Attribut?
- [ ] Gibt es keine „Fake-Links“ (z.B. $\\texttt{\ }$ ohne t e x ttt h re f)?
- [ ] Werden alternative Elemente wie $\\texttt{\ oder $\\texttt{\ }$ mit klarer ARIA-Rolle und t e x ttt t abin d e x verwendet, wenn keine echte Linkfunktion nötig ist?

Schritt 5: Sonderfälle wie Linklisten und Kachel-Teaser

Achte auch bei visuellen Elementen darauf, dass das Linkziel klar ist.

- [ ] Ist der gesamte Titel oder eine informative Beschreibung bei Blog-Teasern verlinkt? (Z.B. „Mehr erfahren: [Titel des Artikels]“)
- [ ] Hat der CTA-Button bei Produktkacheln eine sinnvolle Bezeichnung? (Z.B. „Jetzt kaufen: Produkt 1“)
- [ ] Stellst du sicher, dass bei Pagebuildern das konkrete Call-to-Action-Element verlinkt ist und nicht die gesamte Box ohne klaren Fokus?

Fragen & Antworten

Warum ist Barrierefreiheit bei Links so wichtig, wenn die meisten Leute doch eine Maus benutzen

Auch wenn viele Nutzer eine Maus verwenden, gibt es eine signifikante Anzahl von Menschen, die auf alternative Eingabemethoden angewiesen sind. Das können Menschen mit motorischen Einschränkungen sein, die eine Tastatur oder spezielle Schalter bedienen, oder auch Menschen mit Seheinschränkungen, die Screenreader nutzen. Für diese Gruppen sind korrekt implementierte und klar beschriebene Links absolut entscheidend, um Inhalte überhaupt erreichen und verstehen zu können. Zudem profitieren auch Nutzer ohne Einschränkungen von einer klaren und logischen Linkstruktur – es macht die Navigation für alle einfacher und intuitiver.

Wann sollte ich ein textttaria−label verwenden und wann lieber nicht?

Du solltest ein t e x ttt a r ia − l ab e l nur dann verwenden, wenn der visuelle Linktext nicht ausreicht , um den Zweck des Links für Screenreader-Nutzer klar zu beschreiben. Ein gutes Beispiel ist ein „Bleistift-Icon“ als Link zum Bearbeiten. Optisch ist es klar, aber für einen Screenreader ohne t e x ttt a r ia − l ab e l = „ E in t r a g b e a r b e i t e n „ unverständlich.

Verwende es nicht , wenn der sichtbare Linktext bereits aussagekräftig genug ist. Ein t e x ttt a r ia − l ab e l würde den vorhandenen Text für Screenreader-Nutzer überschreiben und könnte zu Redundanz oder Verwirrung führen. Wenn dein Link bereits „Produktdetails anzeigen“ lautet, ist ein t e x ttt a r ia − l ab e l = „ P ro d u k t d e t ai l s an ze i g e n „ überflüssig und kontraproduktiv, da es den Link unnötig lang macht. Im Zweifelsfall ist es oft besser, den sichtbaren Linktext so klar wie möglich zu formulieren, anstatt auf ein t e x ttt a r ia − l ab e l zurückzugreifen.

Was ist der Unterschied zwischen einem Link und einem Button aus Barrierefreiheits-Sicht?

Obwohl beide klickbar sind, haben sie unterschiedliche semantische Bedeutungen und Rollen für Hilfstechnologien:

- Links ($\\texttt{\ ): Sind dazu da, den Nutzer zu einer neuen Ressource oder einem neuen Ort (intern oder extern) zu navigieren. Sie verändern den „Ort“ des Nutzers im Web.
- Buttons ($\\texttt{\ ): Sind dazu da, eine Aktion auszuführen oder den Zustand der aktuellen Seite zu verändern, ohne eine Navigation zu einer neuen Seite auszulösen (z.B. ein Formular absenden, ein Menü öffnen/schließen, Inhalte filtern, eine Lightbox öffnen).

Die korrekte Verwendung ist wichtig, da Screenreader und andere Hilfsmittel unterschiedliche Erwartungen an Links und Buttons haben. Ein Button, der wie ein Link aussieht, oder umgekehrt, kann verwirrend sein.

Müssen alle Bilder, die als Link dienen, einen Alt-Text haben?

Ja, unbedingt! Wenn ein Bild als Link verwendet wird (z.B. ein Logo, das zur Startseite führt), muss es einen aussagekräftigen t e x ttt a lt -Text haben, der das Linkziel beschreibt. Für Screenreader-Nutzer ist der t e x ttt a lt -Text das Einzige, was ihnen über den Inhalt des Bildes als Link Aufschluss gibt.

Beispiel: $\\texttt{\ \ \}$

Ohne t e x ttt a lt -Text würde der Screenreader möglicherweise nur „Grafik“ oder den Dateinamen vorlesen, was nutzlos wäre. Der t e x ttt a lt -Text sollte hier den Zweck des Links beschreiben, nicht nur den Inhalt des Bildes.

-

Was passiert, wenn ich einen Link habe, der ein Pop-up oder eine Lightbox öffnet? Wie mache ich das barrierefrei?

Wenn ein Link ein Pop-up oder eine Lightbox öffnet, sollte er idealerweise darauf hinweisen. Hier sind zwei Ansätze:

- Informativer Linktext: Beschreibe im Linktext, was passiert.
- Beispiel: $\\texttt{\ AGB in neuem Fenster öffnen\}$
- ARIA-Attribute: Nutze ARIA-Attribute, um den Zweck zu verdeutlichen.
- t e x ttt a r ia − ha s p o p u p = „ d ia l o g „ oder t e x ttt a r ia − ha s p o p u p = „ t r u e „ : Informiert Screenreader, dass nach dem Klick ein Pop-up oder Dialogfeld erscheint.
- t e x ttt a r ia − e x p an d e d = „ f a l se / t r u e „ : Zeigt an, ob das Pop-up aktuell geöffnet oder geschlossen ist.
- t e x ttt a r ia − co n t ro l s und t e x ttt t abin d e x sind auch wichtig, um den Fokus richtig zu steuern und die Bedienung innerhalb des Pop-ups zu ermöglichen. Für komplexere Interaktionen dieser Art solltest du dich aber am besten mit einem erfahrenen Entwickler abstimmen, da die korrekte Implementierung anspruchsvoll sein kann.
-