

Barrierefreie Webseiten

Grundlagen, Anwendung, Praxis

» Hier geht's
direkt
zum Buch

DIE LESEPROBE

Kapitel 2









Bedürfnisse der Nutzenden verstehen, und zwar aller

Kennen Sie Ihre Nutzenden?

Die Themen im Überblick:

- ▶ Menschzentrierter Gestaltungsprozess – die Nutzenden in den Mittelpunkt
- ▶ Heterogene Nutzende – Betroffene integrieren
- ▶ Screenreader und Co. – Unterstützung durch assistive Technologien
- ▶ Design und Barrierefreiheit – ein Widerspruch?
- ▶ Kulturelle Besonderheiten
- ▶ Barrierefreiheit ist für alle ein Gewinn
- ▶ Zusammenfassung des Kapitels

Die kontinuierliche Berücksichtigung von Aspekten der Barrierefreiheit erfordert unter Umständen einen Perspektivenwechsel, ein »Hineinversetzen in die Situation anderer«. Probieren Sie bitte Folgendes mit Ihrer Website aus:

1. Sofern Sie es nicht bereits im letzten Kapitel durchgeführt haben, legen Sie die Maus zur Seite und benutzen Sie ausschließlich die Tastatur. Lässt sich alles per Tastatur steuern? Benutzen Sie die Tabulatortaste (-Taste) und die Pfeiltasten (, ,  und ) zum Navigieren. Mit der -Taste oder der Leertaste (Space) aktivieren Sie die fokussierten Elemente. Mit  (Shift) + der entsprechenden Taste gelangen Sie zum vorherigen Element.
2. Wenn Sie die Webseite ausschließlich in Graustufen darstellen, welchen Eindruck vermittelt sie, wie wirkt sie dann? Simulieren Sie Rot-Grün-Schwäche. Ist alles kontrastreich und unterscheidbar? Verwenden Sie dazu z. B. das kostenlose Silktide-Plug-in (siehe Abschnitt 1.7.2 bzw. Abbildung 2.1).
3. Fortgeschritten: Benutzen Sie einen Screenreader (z. B. VoiceOver für macOS oder NVDA für Windows, siehe Abschnitt 2.3, »Arten von Behinderungen und die Konsequenzen für Ihre Website/Anwendung«). Bewegen Sie sich mit geschlossenen Augen durch die Website (-Taste und Schnellnavigationstasten). Was für Informationen kommen bei Ihnen an? Deckt sich der Informationsgehalt mit dem

Ihnen bekannten visuellen Eindruck? Details zur Nutzung von Screenreadern und eine Schritt-für-Schritt-Anleitung für den kostenlosen Screenreader NVDA finden Sie in Abschnitt 2.3.2, »Rot-Grün-Schwäche bzw. Farbenblindheit«.

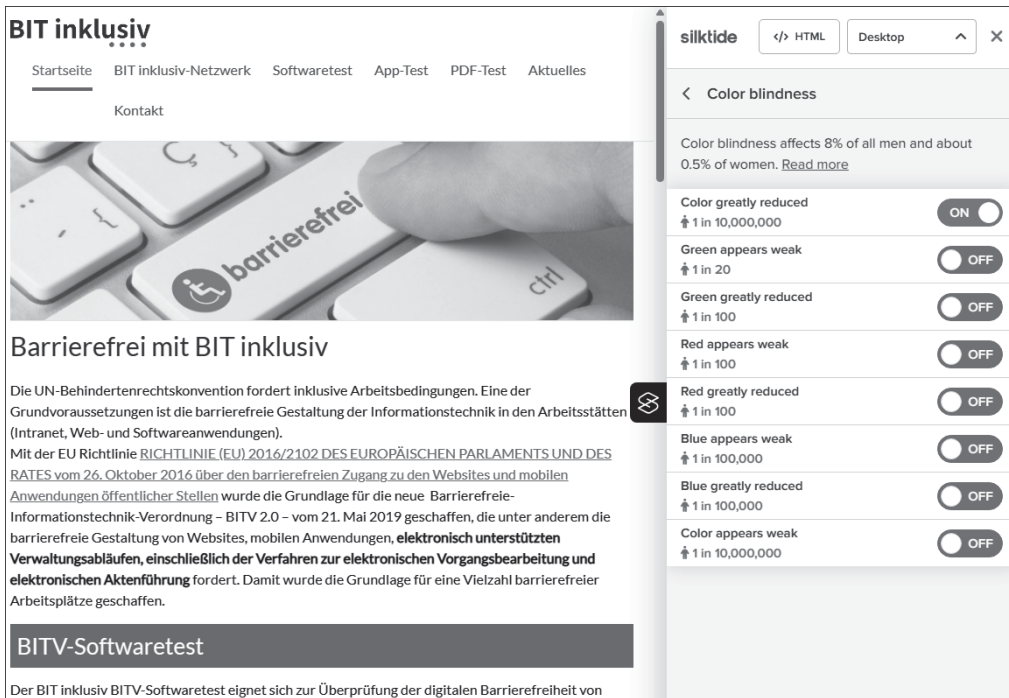


Abbildung 2.1 Simulation von Farbschwächen (Farbenblindheit)

Aus eigener Erfahrung folgender Tipp: Die Sensibilisierung für das Thema Barrierefreiheit ist sehr wichtig. Wenn Nutzende eine Website ausschließlich mit einem Screenreader oder mit einer Simulation einer peripheren Sehstörung nutzen, merken sie erst, welche Hürden sie im Design für die Nutzenden aufgebaut haben. Dass andere Menschen die Website auf eine ganz andere Art und Weise nutzen, haben wir meist »nicht auf dem Schirm«. Sehende Menschen haben alles auf einen Blick, während sehbehinderte Menschen eine Website linear und Stück für Stück erfassen.

2.1 Menschzentrierter Gestaltungsprozess – die Nutzenden in den Mittelpunkt stellen

Bei der Webseitenoptimierung wird sehr häufig auf Personas oder Benutzergruppen zurückgegriffen und versucht, deren Bedürfnisse, Sorgen und Nöte kontinuierlich im Design zu berücksichtigen. Möglicherweise setzen auch Sie Personas in Ihrem Unter-

nehmen ein. Hand aufs Herz: haben Sie dabei auch Aspekte der Barrierefreiheit berücksichtigt?

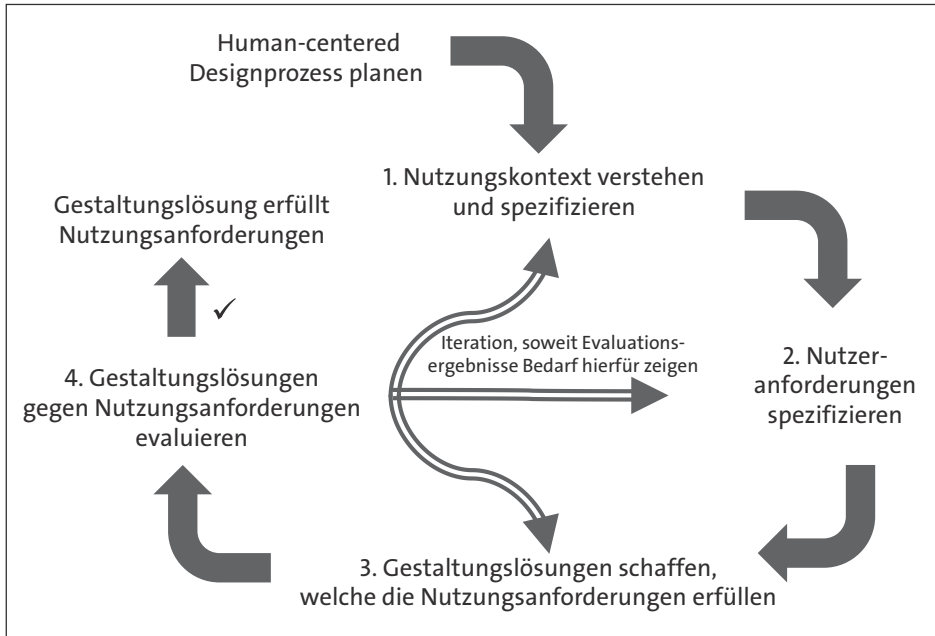


Abbildung 2.2 Menschzentrierte Gestaltung

In der Praxis wird sehr häufig der Ansatz der menschzentrierten Gestaltung (auch als Human-centered Design bekannt) gewählt. Dabei geht es darum, Produkte, interaktive Systeme und Dienstleistungen so zu gestalten, dass sie den Bedürfnissen, Fähigkeiten und Einschränkungen der Nutzenden entsprechen. Ziel ist es, die Benutzerfreundlichkeit und -zufriedenheit zu maximieren, indem der Mensch in den Mittelpunkt des Designprozesses gestellt wird. Durch wiederholte Tests und Anpassungen wird das Produkt schrittweise verbessert. Die Nutzenden werden kontinuierlich in den Gestaltungsprozess einbezogen, um sicherzustellen, dass ihre Anforderungen und Erwartungen verstanden und berücksichtigt werden. Die DIN EN ISO 9241 ist eine international anerkannte Normenreihe, die Richtlinien zur Ergonomie von Mensch-System-Interaktionen gibt. Teil 210 dieser Norm, »Menschzentrierte Gestaltung interaktiver Systeme«, beschreibt die vier Phasen der menschzentrierten Gestaltung interaktiver Systeme (siehe Abbildung 2.2):

1. Nutzungskontext verstehen und beschreiben: Untersuchung der Nutzenden, ihrer Aufgaben und des Nutzungskontextes, d. h.:
 - Wer sind meine Nutzenden?
 - Welche Aufgaben soll das System erfüllen?

- In welcher Umgebung wird das System eingesetzt?
 - Welche Ressourcen stehen zur Verfügung?
 - Was können meine Nutzenden, was nicht?
2. Ermittlung von Anforderungen, die sich aus dem Nutzungskontext unter Berücksichtigung der Bedürfnisse und Erwartungen der Nutzenden ergeben, d. h.:
 - Was können die Nutzenden am System eingeben?
 - Was sollen die Nutzenden erkennen können?
 - etc.
 3. Gestaltungslösungen entwickeln: Erstellung von Prototypen und Gestaltungslösungen auf Basis der festgelegten Anforderungen, d. h., wie setzt das Interface-Design die Nutzungsanforderung konkret um?
 4. Gestaltungslösungen evaluieren: Durchführung von Nutzerevaluationen und Tests, zur Überprüfung der Gebrauchstauglichkeit, d. h., erfüllt das Design die Nutzungsanforderungen?

Bei der barrierefreien Gestaltung stehen ebenfalls die Nutzenden im Mittelpunkt (menschzentrierte Gestaltung). Sie sollten also verstehen und konkret wissen, wer die Menschen sind, für die Sie Ihre Website erstellen. Sie sollten wissen, welche Barrieren für Menschen mit Behinderungen auftreten können, damit Sie diese proaktiv vermeiden können.

Barrierefreie Websites sind für alle Menschen von Vorteil, da es temporäre oder situative Einschränkungen gibt die in jeder Lebenslage oder Situation auftreten können, wie z. B., dass ein farbiges Diagramm in Schwarzweiß gedruckt wird. Weitere Beispiele für situative Einschränkungen sind die Nutzung eines Bildschirms in der prallen Sonne oder die Nutzung des Smartphones in einer lauten Umgebung. Beiträge in sozialen Medien werden im Büro oder in einer Vorlesung (heimlich) angesehen, ohne den Ton einzuschalten. Hier kommt die ursprünglich für hörbehinderte Menschen gedachte Transkription von Videos (positiv) zum Tragen. Daher sollten Sie Aspekte der Barrierefreiheit in den menschzentrierten Gestaltungsprozess integrieren.

Tipp: Simulation von Accessibility-Persona

Funkify (<https://www.funkify.org/>) bietet die Möglichkeit, eine Website aus der Sicht verschiedener Personas zu betrachten. Sunshine Sue simuliert Sonnenlicht (siehe Abbildung 2.3), Elderly Ellen ist sehbehindert und zittert, wodurch die Maus bei der Bedienung hin und her springt. Gerade zur Sensibilisierung für das Thema »digitale Barrierefreiheit« ist das Tool ein sehr guter Einstieg.



Abbildung 2.3 Simulation von Sonneneinstrahlung auf einen Bildschirm mithilfe von Funkify

2.2 Heterogene Nutzende – Betroffene integrieren

Werden die Produkte oder Dienstleistungen, die Sie anbieten oder verkaufen, auch für Sie erstellt, d. h., sind Sie der Konsument Ihrer eigenen Produkte? In der Regel sind Sie das nicht! Ein Beispiel aus der Praxis: Bei einem Review der THWS-Seite war bei mehreren der Testenden ein anderes Design »Standard«, das nicht dem »normalen« Standarddesign entsprach. Abbildung 2.4 zeigt das erwartete Design der THWS-Seite mit dunkler Schrift auf hellem Hintergrund.

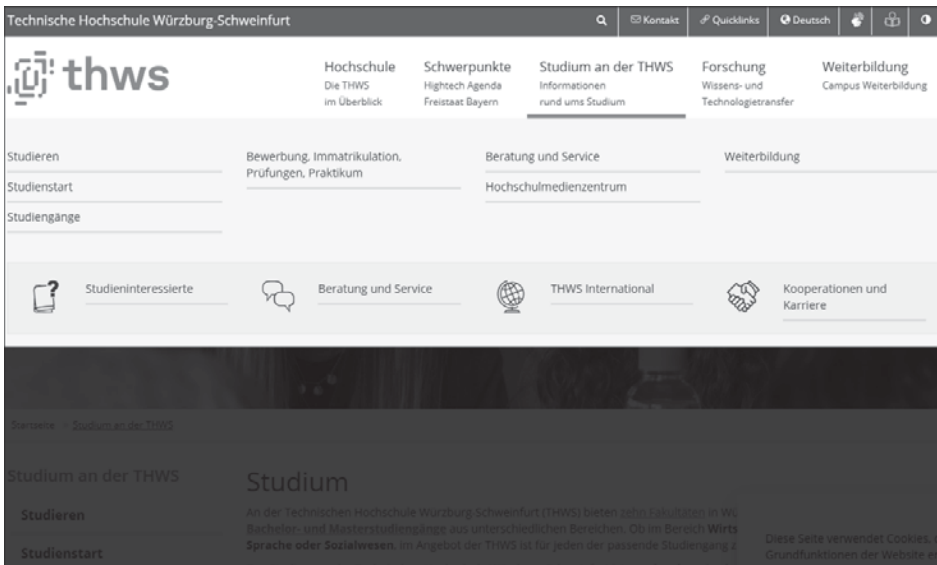


Abbildung 2.4 Design der Website im Standard (kein Dark Mode)

Letztendlich war dies darauf zurückzuführen, dass mehrere Testende den Dark Mode bei Webseiten standardmäßig per Plug-in aktiviert haben, um ihre Augen zu schonen. Abbildung 2.5 zeigt den Dark Mode der Website mit heller Schrift auf dunklem Hintergrund.

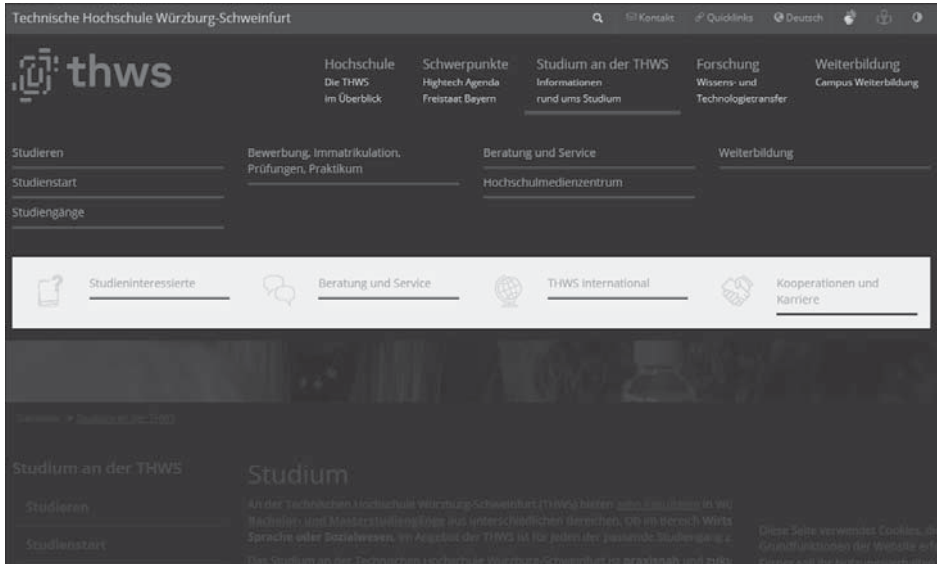


Abbildung 2.5 Darstellung der Website im Dark Mode

Was für Website-Betreibende oder Designende der vermeintliche Normalzustand ist (kein Dark Mode), ist für eine Vielzahl an Nutzenden keineswegs der Standard. Daher lautet unsere klare Empfehlung an Sie, führen Sie User-Tests mit einem breiten Spektrum an Testenden durch. Heterogene Nutzergruppen bieten einen sehr großen Mehrwert für die Sicherstellung der Zugänglichkeit der Website/App. Seit der Corona-Pandemie sind viele Menschen bestens vertraut mit Programmen zur Bildschirmübertragung (Microsoft Teams, Zoom etc.). Das heißt, Sie können von jedem Ort der Welt aus Menschen dabei zusehen, wie sie Ihre Website bzw. Applikation nutzen und wie diese in ihrem Set-up, ihrer individuellen Umgebung und mit ihrer technischen Ausstattung tatsächlich dargestellt wird.

Im Kontext der Usability ist im Sinne der menschenzentrierten Gestaltung (DIN ISO 9241-210) die Beteiligung von Nutzenden während des gesamten Design- und Entwicklungsprozesses eine Erfolg versprechende Grundvoraussetzung. Insbesondere die verwendete technische Ausstattung (Screenreader, Eingabegeräte) sowie behinderungsspezifische Besonderheiten sprechen für individuelle User-Tests. Wer schon einmal (externen) Nutzenden der eigenen Website über die Schulter geschaut hat, wird festgestellt haben, dass das Verhalten sehr individuell ist. Individualisierte Gerä-

te, Verhaltensmuster, mentale Modelle etc. führen auf unterschiedlichen Wegen zu gleichen Ergebnissen oder manchmal auch zu unlösbaren Konflikten und daraus resultierenden Abbrüchen.

Selbsttests auf Basis des BIK BITV-Tests sowie Tools wie Google Lighthouse und andere Browser-Plug-ins unterstützen die Überprüfung von Webanwendungen auf Barrierefreiheit. Ein Großteil des Optimierungspotenzials kann damit bereits in den Phasen des Entwicklungsprozesses maschinell erkannt werden, dennoch ist ein begleitendes manuelles Testen mit Betroffenen sehr zu empfehlen. Das heißt, im ersten Schritt kann eine Analyse mit maschinellen Hilfsmitteln und ein strukturierter Test durchgeführt werden, bevor die Anwendung bzw. Website von einer tatsächlich betroffenen Person mit assistiven Technologien wie einem Screenreader getestet werden sollte. Angebote hierzu bieten diverse Organisationen an, wie z. B. die Stiftung Pfennigparade.

Laut einer internationalen Studie aus dem Jahr 2021 sind 29,1 % der Testenden auf Barrierefreiheit selbst von einer Behinderung (<https://webaim.org/projects/practitionersurvey3/>) betroffen. Gegenüber der vergleichbaren Studie aus dem Jahr 2014 mit 21,8 % ist dies eine bemerkenswerte Steigerung des Anteils von Menschen mit Behinderungen unter den Testenden. Die besten Testergebnisse werden erzielt, wenn die Zielgruppe die Website, App oder Software mit den entsprechenden assistiven Technologien testet. Daher die selbstkritische Frage an Sie: Wie viele Menschen mit Behinderungen haben Ihre Website auf Barrierefreiheit getestet? Wenn Sie getestet haben, wie viele Entwicklungszyklen sind seitdem vergangen?

Was für Usability und Conversion-Optimierung gilt, gilt auch hier: Sie, wir alle sind betriebsblind. Sie können nicht so agieren, als wäre die Website neu. Sie kennen Ihre eigene Website, sehen die Inhalte täglich, verstehen die internen Abkürzungen und wissen, was die Links und Funktionen bewirken. Um möglichst viele Optimierungspotenziale aufzudecken – jede Website hat Optimierungspotenzial, wirklich jede – sollte die Qualitätssicherung in der Entwicklungsabteilung um Aspekte der Barrierefreiheit ergänzt werden und idealerweise mit echten Nutzenden getestet werden. Sofern diese aus der direkten Zielgruppe kommen, wäre dies der Idealfall. Beobachten Sie z. B. blinde Nutzende bei der Bedienung einer Website, so werden Sie im ersten Schritt wahrscheinlich von der schnellen Vorlesegeschwindigkeit der Texte überrascht sein. Da die Navigation für blinde Nutzende eindimensional ist und eine Webseite Überschrift für Überschrift bzw. Bereich für Bereich erfasst wird, ist ein hohes Maß an Aufmerksamkeit und Übung erforderlich.

Neue, unbekannte Nutzende haben eine völlig andere Wahrnehmung von Inhalten, da beim ersten Besuch einfach alles neu ist und nur auf das Wissen aus der Nutzung anderer Websites zurückgegriffen werden kann. Jakob Nielsen hat bereits im Jahr 2000 darauf hingewiesen, dass die Nutzenden die meiste Zeit auf anderen Websites verbringen und daher im Idealfall auf Gelerntes und gewohnte Konventionen

zurückgegriffen werden sollte (Jakob's Law of the Internet User Experience, <https://www.nngroup.com/articles/end-of-web-design>). Steve Krug drückte es in seinem gleichnamigen Buch noch einfacher aus:

»Don't make me think!«

Ergänzen Sie daher klassische Usability-Tests um Tests der Barrierefreiheit. Vergleichbar mit verschiedenen Kundengruppen (Persona) sollten verschiedene Behinderungen bzw. Anforderungen getestet werden. Remote Testing, bei dem die Testenden das individuelle Set-up (Eingabegerät, Screenreader etc.) nutzen können, spart Zeit und Geld und erhöht gleichzeitig die Effektivität und Authentizität. Kooperationen (z. B. mit lokalen Einrichtungen wie Blindenschulen) und/oder Barrierefreiheit-Beratende ergänzen den (bestehenden) Usability-Testpool.

2.3 Screenreader und Co. – Unterstützung durch assistive Technologien

Sie haben nun schon mehrfach von Screenreadern und anderen assistiven Technologien gelesen. Dieser Abschnitt soll Ihnen einen Überblick über assistive Technologien geben und einen Einstieg zur Nutzung eines Screenreaders aufzeigen.

2.3.1 Was sind assistive Technologien?

Assistive Technologien sind Hardware- und Softwarelösungen, die Menschen mit Behinderungen ermöglichen, Zugang zu digitalen Informationen (Website, App, PDF etc.) zu erhalten. Im Bereich der digitalen Barrierefreiheit gibt es eine breite Palette von assistiven Technologien, wie z. B.:






- ▶ Screenreader: Software, die Bildschirminhalte in Sprache umwandelt, um sehbehinderten oder blinden Menschen digitale Informationen zugänglich zu machen, Beispiele: JAWS, NVDA, VoiceOver
- ▶ Braillezeilen: Geräte, die Bildschirmtext in Braille (Blindenschrift) umwandeln und über eine taktile Oberfläche ausgeben
- ▶ Vergrößerungssoftware: Software, die Text oder andere Objekte auf dem Bildschirm vergrößert, um die Lesbarkeit für Menschen mit eingeschränktem Sehvermögen zu verbessern, Beispiele: ZoomText, Windows Magnifier
- ▶ Untertiteldienste: verfügbar für Videos oder Livestreams, entweder durch automatische Transkription (z. B. bei YouTube) oder durch manuell erstellte Untertitel

- ▶ Sprachsteuerungssysteme: Ermöglichen die Steuerung von Geräten (Computer, Smartphones) durch gesprochene Befehle, Beispiele: Siri, Google Assistant, Dragon NaturallySpeaking.
- ▶ Alternative Eingabegeräte: Geräte, die herkömmliche Tastaturen und Mäuse ersetzen, wie z. B. Eyetracking, Mund-Maus-Systeme (Sip-and-Puff-Geräte) oder Joysticks
- ▶ Anpassbare Tastaturen und Mäuse: Diese speziellen Eingabegeräte haben größere Tasten oder ein ergonomisches Design, um das Tippen oder Navigieren zu erleichtern.
- ▶ Text-to-Speech-Systeme (TTS: Software, die geschriebenen Text vorliest, um Menschen mit Leseschwierigkeiten oder Lernbehinderungen zu helfen.

Weitere Details und Beispielbilder finden Sie unter <https://webaim.org/articles/motor/assistive>.

2.3.2 Einrichten des kostenlosen Screenreaders NVDA

Screenreader sind elementare assistive Technologien, die Bildschirminhalte vorlesen oder eine taktile Ausgabe über eine angeschlossene Braillezeile (Blindenschrift) ermöglichen. Neben kommerziellen Lösungen wie JAWS (Job Access With Speech) gibt es kostenfreie Open-Source-Lösungen wie NVDA (NonVisual Desktop Access). Die Hersteller von Betriebssystemen bieten mit VoiceOver (Apple), Narrator (Microsoft Windows) oder Google für Android mit TalkBack integrierte Möglichkeiten zur Sprachausgabe und Interaktion. Für Tests bzw. Optimierungen während der Entwicklung sind diese kostenlosen Möglichkeiten ausreichend. Laut einer Studie von Webaim vom Januar 2024 verwenden 91,3 % der Screenreader-Nutzenden auch ein mobiles Endgerät (<https://webaim.org/projects/screenreadersurvey10/#used>).

In vielen Usability-Sitzungen ist zu beobachten, dass Formulare bevorzugt mit der -Taste anstelle mit der Maus ausgefüllt werden, weil es schneller geht. Die Eingabe von Befehlen über die Tastatur ermöglicht es auch sehenden Menschen, viel schneller durch Anwendungen oder Webseiten zu navigieren. Mit der Tastenkombination  +  in Excel wird die aktuelle Zelle kopiert und mit  +  eingefügt. Die Nutzung über die Tastenkombination ist schneller als die entsprechende Eingabe über die Maus. Nicht nur aus Gründen der Einfachheit, sondern auch aus Gründen der Notwendigkeit für blinde Menschen sollte die Webseite auch ausschließlich über Tastatureingaben bedienbar sein. So wie viele Menschen lieber ein Hörbuch konsumieren, als ein Buch auf Papier zu lesen, so nutzen viele Menschen lieber die Möglichkeit der Sprachausgabe, als eine Webseite zu lesen. Weiterhin kann die

Sprachausgabe von Menschen genutzt werden, die nicht oder nur eingeschränkt lesen können (insbesondere Analphabeten) oder wenn Aufgaben parallel erledigt werden (z. B. Lesen längerer Texte und gleichzeitiges Betrachten von Bildern). Die Einsatzmöglichkeiten von Screenreadern sind deshalb vielfältiger Natur.

Den kostenlosen und leistungsstarken Screenreader NVDA können Sie unter <https://www.nvaccess.org/download/> für Microsoft Windows herunterladen, er ist ohne weitere Registrierung voll nutzbar. Wenn Sie einen Mac verwenden, empfehlen wir Ihnen, sich eine virtuelle Maschine (z. B. über Parallels Desktop for Mac) einzurichten oder stattdessen VoiceOver zu benutzen.

Bevor Sie mit der Sprachausgabe beginnen, ist es wichtig, eine **NVDA**-Taste (steht hier stellvertretend für Ihre gewählte Taste) zu definieren. Sie können in den Optionen nicht nur die Sprechstimme, Geschwindigkeit etc. definieren, sondern auch die **NVDA**-Taste. Viele Befehle werden über die **NVDA**-Taste in Kombination mit einer anderen Taste aufgerufen. In den meisten Fällen ist dies die **Einfüg**-Taste oder die **⇧**-Taste (Feststelltaste). Abbildung 2.6 zeigt die Optionen von NVDA mit der Festlegung der **NVDA**-Taste.

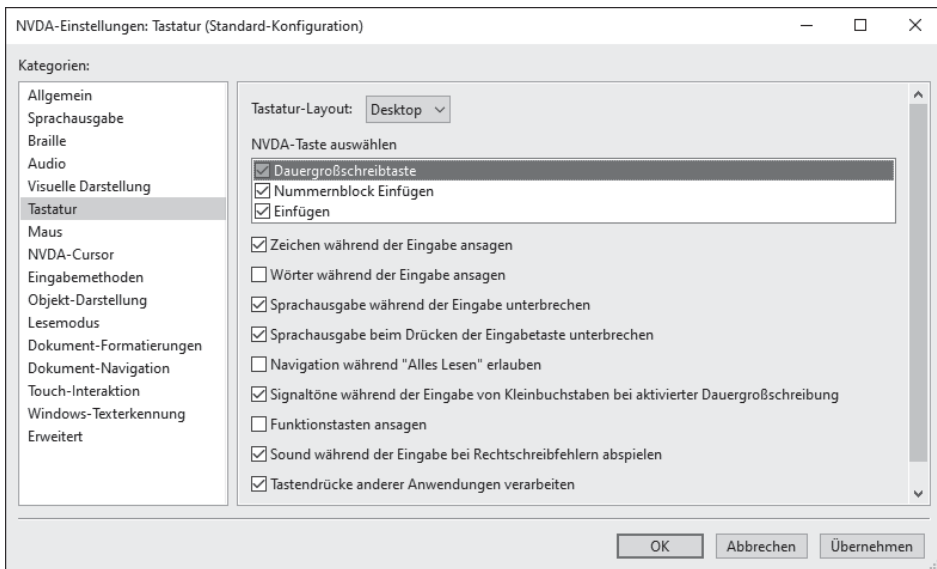


Abbildung 2.6 Definition der NVDA-Taste

Mit der Tastenkombination **Strg** + **Alt** + **N** können Sie z. B. den Screenreader starten und mit **NVDA** + **Q** beenden. Wenn Sie die **NVDA**-Taste als Feststelltaste definiert haben, wäre dies die Tastenkombination **⇧**-Taste + **Q**.

Tabelle 2.1 zeigt die Tastenkombinationen für eine erste effiziente Nutzung. Durch Drücken der **Strg**-Taste können Sie die Sprachausgabe jederzeit temporär stoppen.

Funktion	Tastenkombination
Screenreader starten	Strg + Alt + N
Screenreader beenden	NVDA + Q
Sprachmodus ändern	NVDA + S
fokussierbares Element	↔
vorheriges Element	⏪ + ↔ Die Kombination mit der ⏪ -Taste führt jeweils zum vorherigen Element.
fokussiertes Element auswählen	↵ (Return/Enter)
Elementliste anzeigen	NVDA + F7
Regionen (Landmarks)	D
Überschriften	H
Überschrift H1–H6	1 – 6
Formular	F
Liste	L
Link	K
Tabellen	T

Tabelle 2.1 Tastaturbefehle in NVDA

Hinweis: Tastaturbefehle für JAWS

Es gibt verschiedene Listen mit Tasturbefehlen – nicht nur für NVDA, sondern auch für JAWS. Eine sehr gute Übersicht für JAWS bietet <https://dequeuniversity.com/screenreaders/jaws-keyboard-shortcuts>.

Starten Sie NVDA und verschaffen Sie sich einen ersten Überblick, indem Sie mit der Tabulatortaste (**↔**) durch Ihre Webseite navigieren. Mit der **↵**-Taste (Return) können Sie die Links aktivieren und so von Seite zu Seite navigieren. Einen ersten Über-

blick über den Inhalt der Webseite erhalten Sie, wenn Sie nacheinander die Taste **[H]** drücken. Dadurch wird Ihnen die erste Überschrift vorgelesen. Bei nochmaligem Drücken die nächste Überschrift, usw.

Mit den Tasten **[D]**, **[F]**, **[L]**, **[K]** und **[T]** können Sie direkt die spezifischen Bereiche anspringen bzw. diese auswählen. Zusätzlich zur Sprachausgabe können Sie im NVDA-Menü über **WERKZEUGE • SPRACHAUSGABEN-BETRACHTER** eine Textausgabe erzeugen. Dadurch können Sie visuell prüfen, was den Nutzenden vorgelesen wird (ein Beispiel für eine Textausgabe zeigt Abbildung 2.7).

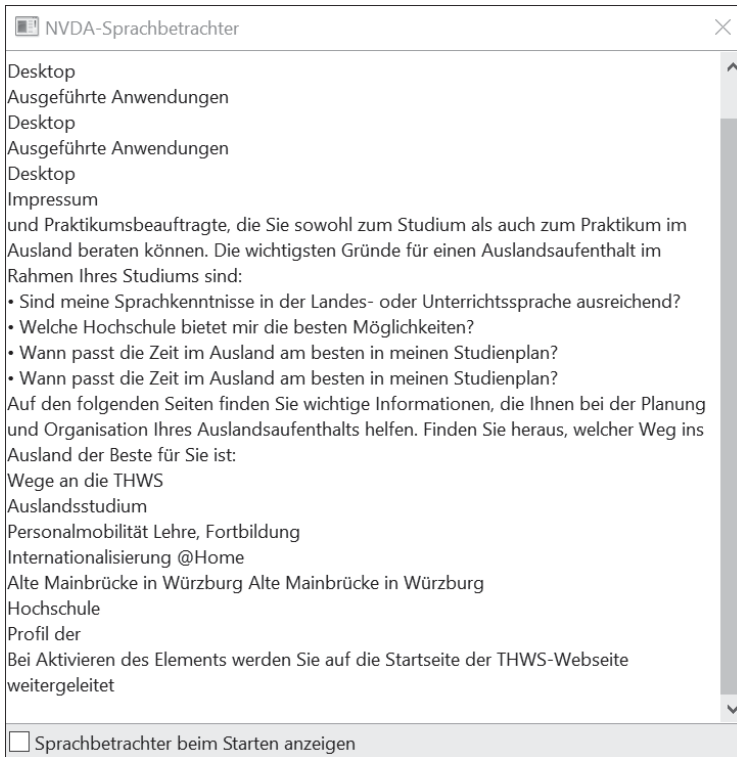


Abbildung 2.7 NVDA-Sprachbetrachter

Die Elementliste (unter anderem Links, Formulare, Überschriften auf einer Webseite) kann auch über die Tastenkombination **[NVDA]-Taste + [F7]** aufgerufen werden. Über diese Liste erhalten Sie einen schnellen Überblick, was NVDA als Überschrift in welcher Reihenfolge erkennt. Weiterhin werden die auf der Webseite vorhandenen Formularfelder und Links angezeigt. Für die Fehlerprüfung ist dies eine hilfreiche Übersicht. Abbildung 2.8 zeigt eine beispielhafte Übersicht der (erkannten) Überschriften einer Webseite.

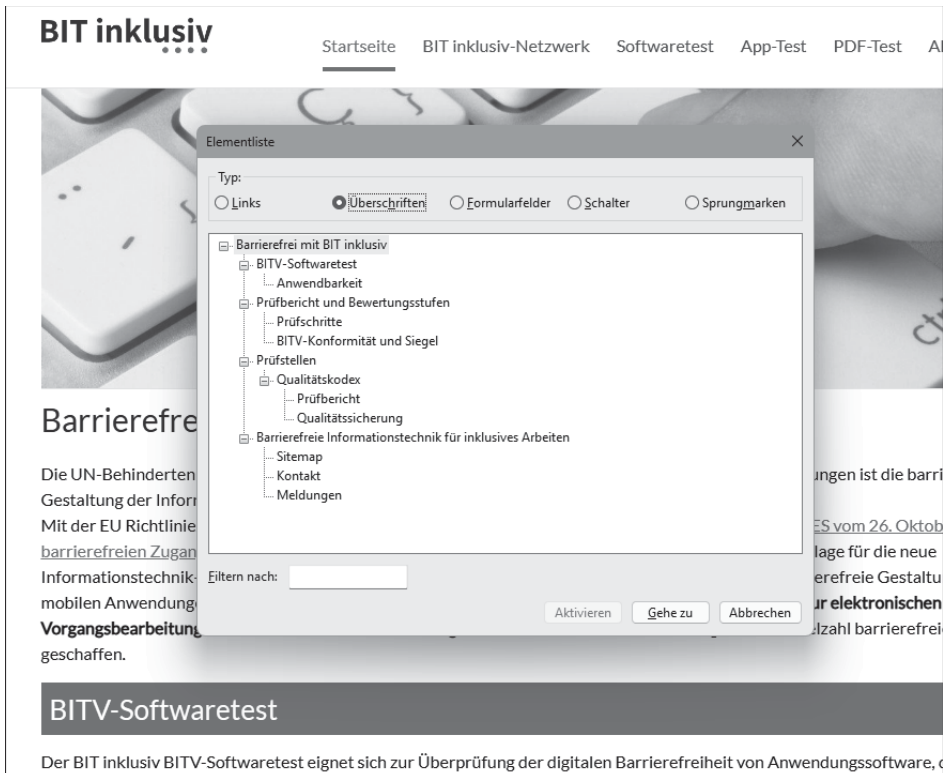


Abbildung 2.8 Erkannte Elemente auf der Seite – die Elementliste

2.4 Design und Barrierefreiheit – ein Widerspruch?

Um die Frage gleich zu beantworten: Nein, Design und Barrierefreiheit widersprechen sich nicht. Oder finden Sie die Apple-Website unästhetisch? In der heutigen digitalen Welt ist Barrierefreiheit nicht nur eine rechtliche Verpflichtung, sondern auch ein wichtiger Aspekt bei der Gestaltung benutzerfreundlicher Anwendungen und Websites. Barrierefreiheit und User Experience (UX) sind eng miteinander verbunden und beeinflussen sich gegenseitig. Eine nutzerfreundliche Anwendung reduziert Frustrationen und Fehler und erhöht damit die Effizienz und Effektivität. Langfristig fördert eine gute UX die Kundenbindung. Usability ist bereits per Definition ein zentraler Aspekt der User Experience. Eine vermeintlich positive UX ohne Berücksichtigung der Barrierefreiheit ist unvollständig. Eine Anwendung, die vollständig mit der Tastatur bedienbar ist, unterstützt Nutzende mit motorischen Einschränkungen (Barrierefreiheit), verbessert die Effizienz der Bedienung (Usability) und erhöht die Zufriedenheit (UX).

Das folgende Beispiel soll die Komponenten von Usability nochmals näher beleuchten. Effektivität misst, inwieweit Nutzende ihre Ziele vollständig und richtig erreichen können, d. h. ihre beabsichtigten Aufgaben erfolgreich und vollständig abschließen können. Möchten Sie z. B. in einem Onlineshop eine Bestellung stornieren und bietet der Shop diese Möglichkeit an, so ist dies für Sie effektiv. Effizienz beschreibt dabei, wie ressourcenschonend Sie Ihr Ziel erreichen können (typischerweise mit wenig Zeit und Aufwand). Zufriedenheit bezieht sich auf das subjektive Empfinden, d. h., wie angenehm und befriedigend die Nutzung des Systems für Sie ist: »Das war einfach«, »Das war intuitiv«, »Mit der Hilfe bin ich zufrieden«.

Vorteile der Kombination von UX und Barrierefreiheit:

- ▶ Barrierefreiheit öffnet Ihre Anwendung für einen größeren Nutzerkreis, einschließlich der oft zahlenmäßig unterschätzten Gruppe der Menschen mit Behinderungen.
- ▶ Barrierefreie Designs sind in der Regel klarer und intuitiver, was die Usability für alle Nutzenden erhöht. Die Website von Apple ist aus Sicht der Barrierefreiheit als sehr gut zu bewerten und zeigt, dass Barrierefreiheit, ansprechendes Design und UX Hand in Hand gehen können.
- ▶ Durch die Integration von Barrierefreiheit in die UX stellen Sie sicher, dass Ihre Anwendung den gesetzlichen Anforderungen entspricht.
- ▶ Eine barrierefreie und benutzerfreundliche Anwendung fördert ein positives Image und erhöht die Kundenbindung.

Nachteile und Herausforderungen:

- ▶ Die Umsetzung eines barrierefreien Designs kann zusätzlichen Zeit- und Kostenaufwand erfordern, insbesondere dann, wenn sie nicht von Anfang an berücksichtigt wird.
- ▶ Die (nachträgliche) Integration von Barrierefreiheit in bestehende Systeme kann technisch anspruchsvoll sein.
- ▶ Ein häufiges Argument gegen barrierefreies Design ist, dass es ästhetische Einschränkungen mit sich bringen kann. Designende haben oft das Gefühl, dass die Notwendigkeit, kontrastreiche Farben, größere Schriften und klare, einfache Layouts zu verwenden, die kreativen Möglichkeiten einschränkt. Die Notwendigkeit, sich mit verschiedenen Technologien, Tools und Standards zur Barrierefreiheit vertraut zu machen, kann als negativ empfunden werden.

Die Integration von Barrierefreiheit in die Usability und UX-Strategie ist nicht nur eine ethische Verpflichtung, sondern auch eine Möglichkeit, die Benutzerfreundlichkeit und das Gesamterlebnis aller Nutzenden zu verbessern. Die Kombination dieser Konzepte kann zu inklusiven, effizienten und zufriedenstellenden Produkten führen. Es ist jedoch wichtig, die potenziellen Herausforderungen und Kompromisse zu

erkennen und anzugehen, um eine Balance zwischen ästhetischem Design und Barrierefreiheit zu finden.

2.5 Kulturelle Besonderheiten

Wie interpretieren Sie das Zeichen »Daumen hoch«? Positiv? In Deutschland und anderen Ländern bedeutet es »alles okay, gut gemacht«, im Irak und Iran z. B. gilt es als obszöne Geste. Kultur, individueller Hintergrund und Vorlieben können die Wahrnehmung und Nutzung von Elementen einer Website oder App erheblich beeinflussen. Kulturelle Unterschiede spielen eine bedeutende Rolle bei der Gestaltung barrierefreier Lösungen. So haben Farben und visuelle Symbole wie das obige »Daumen hoch« in verschiedenen Kulturen unterschiedliche Bedeutungen. Ein Design, das in einer Kultur als positiv und ansprechend empfunden wird, kann in einer anderen Kultur als unangemessen oder sogar beleidigend angesehen werden. In unserer westlichen Kultur wird die Farbe Rot oft mit Warnung oder Fehler assoziiert, während sie in einigen asiatischen Kulturen Glück und Wohlstand symbolisiert.

Barrierefreiheit muss daher nicht nur sprachliche Unterschiede und Schriftarten, sondern auch kulturelle Besonderheiten berücksichtigen. Einige Sprachen verwenden komplexe Schriftzeichen, die spezielle Schriftarten und größere Schriftgrößen erfordern. Chinesische Schriftzeichen benötigen mehr Platz und Deutlichkeit als lateinische Buchstaben, was bei der Gestaltung von Textinhalten berücksichtigt werden muss. Die Leserichtung variiert von Kultur zu Kultur. Während in westlichen Kulturen von links nach rechts gelesen wird, lesen Menschen in arabischen und einigen asiatischen Kulturen von rechts nach links. Diese Unterschiede müssen bei der Gestaltung von Benutzeroberflächen und der Platzierung von Navigationselementen berücksichtigt werden.

Menschen mit Migrationshintergrund können aufgrund ihres kulturellen und sprachlichen Hintergrundes spezifische Bedürfnisse und Erwartungen an die Barrierefreiheit haben. Viele Menschen mit Migrationshintergrund beherrschen ihre Muttersprache nicht vollständig. Mehr als die Hälfte (ca. 54 %) der Menschen mit Einwanderungsgeschichte nutzten neben Deutsch mindestens eine weitere Sprache zur Verständigung mit Haushaltsmitgliedern (Daten von 2022). Weitere knapp 23 % sprechen zu Hause ausschließlich eine oder mehrere andere Sprachen als Deutsch.¹ Um die Barrierefreiheit zu gewährleisten, sind eine mehrsprachige Unterstützung oder/und eine klare und einfache Sprache unerlässlich. Die Darstellung der Inhalte sollte kultursensibel sein und keine Stereotype oder Vorurteile fördern. Das heißt, die persönlichen Vorlieben und Bedürfnisse spielen eine zentrale Rolle bei der Wahrnehmung und Nutzung barrierefreier Lösungen. Menschen haben unterschiedliche Prä-

¹ https://www.destatis.de/DE/Presse/Pressemitteilungen/2024/02/PD24_NO08_12.html

ferenzen in Bezug auf die Darstellung von Informationen und die Bedienung digitaler Produkte. Die Möglichkeit, Benutzeroberflächen anzupassen, kann die Barrierefreiheit erheblich verbessern. Die Nutzenden sollten daher in die Lage versetzt werden, z. B. Schriftgrößen, Farben und Kontraste ihren individuellen Bedürfnissen anzupassen.

2.6 Barrierefreiheit ist für alle ein Gewinn

Im Bereich der Suchmaschinenoptimierung wird auf Updates in den Algorithmen der Suchmaschinen reagiert bzw. kontinuierlich an technischen Optimierungen gearbeitet, um bessere Rankings zu erzielen. Gleichzeitig führen die Unternehmen permanent A/B-Test durch, um die Konversionsrate (Conversion Rate) der Website zu erhöhen und letztlich höheren Umsatz zu erzielen. Neben der Umsatzsteigerung ist auch der Aspekt der Kostensenkung von Bedeutung. Gerade im Kundenservice können durch barrierefreie Websites Kosten eingespart werden, da Nutzende z. B. die Informationen umfassender selbst abfragen können (und deshalb nicht mehr anrufen müssen). Weitere Kostenoptimierungen könnten sich durch eine geringere Retourenquote ergeben, da sich die Nutzende intensiver mit den Informationen zum Produkt oder zur Dienstleistung auseinandersetzen konnten. Und langfristig betrachtet kommt ein loyaler Wiederkäufer deutlich günstiger als die Akquisition eines Neukunden.

Zur weiteren Veranschaulichung folgt in Abbildung 2.9 ein Rechenbeispiel, was Barrierefreiheit an zusätzlichem Umsatz bzw. Deckungsbeitrag ermöglicht. Durch die Optimierung der Barrierefreiheit (z. B. Tastaturnutzung), die Reduzierung von Absprüngen und die verbesserte Usability ergibt sich in dem Beispiel eine Steigerung der Konversionsrate um 20 %. Da die Werbekosten konstant bleiben und nur mehr Nutzende konvertieren, erhöht sich der Deckungsbeitrag nicht nur um 20 %, sondern sogar um 36 %.

Eine weitere Fokussierung auf digitale Barrierefreiheit unterstützt die Suchmaschinenoptimierung durch technische Optimierungen in HTML bzw. CSS. Weiterhin werden positive Nutzersignale durch intensivere Nutzung der Webseiten an die Suchmaschinen vermittelt. Betrachten Sie das zusätzliche Zielgruppenpotenzial, das sich durch universelles, barrierefreies Design ergibt, eröffnen sich für Sie weitere Umsatz- bzw. Marktpotenziale. Zudem profitieren alle Menschen von barrierefreien Websites, also insgesamt eine Win-win-Situation. Dennoch erfordert die Bereitstellung barrierefreier Websites und Apps ein hohes Maß an Planung sowie eine zielgerichtete und stringente Umsetzung. Dies ist unweigerlich mit Zeit und Kosten verbunden. Es ist fraglich, ob eine Website bei der heutigen Dynamik der Webentwicklung und ständigen Anpassung von Inhalten dauerhaft zu 100 % barrierefrei sein kann. Suchmaschi-

nenoptimierung und Konversionsoptimierung sind nie abgeschlossen, es wird keinen finalen Zustand geben. Gleiches gilt für die digitale Barrierefreiheit, auch hier können neue Anforderungen jederzeit proaktiv zu zufriedenen Nutzenden führen. Daher ist eine kontinuierliche Anpassung bzw. Optimierung von großer Bedeutung. Ähnlich wie Konversionsoptimierung und Suchmaschinenoptimierung ist digitale Barrierefreiheit kein Projekt mit einem definierten Anfang und Ende, sondern ein kontinuierlicher Prozess. Dabei schließen sich Barrierefreiheit und ansprechendes Design keineswegs aus – die Website von Apple.com ist dafür ein hervorragendes Beispiel mit sehr guter Zugänglichkeit.

Beispielrechnung Barrierefreiheit

Umsatz (User) / Tag	10.000 €
Verlust durch Conversion	3.000 €
Konversionsrate	1,5%
Besuchungen / Tag	100
potenzieller Umsatz / Tag	7.000,00 €
Werbekosten	2.000,00 €
weitere variable Kosten (FK)	2.000,00 €
Besuchungskosten / Tag	2.000,00 €
bei 20% Absatzen	600.000,00 €

Optimierung der Konversionsrate, d.h. ein weiteres Anheben der Zahl von Konversionen	
neue Konversionsrate	1,80%
Umsatzsteigerung / Tag	1.800,00 €
	88.000 €
Werbekosten	2.000,00 €
weitere variable Kosten (FK)	2.000,00 €
Besuchungskosten / Tag	5.400,00 €
	82.000 €
bei 20% Absatzen	600.000,00 €

Alle anderen Werte bleiben

Abbildung 2.9 Beispielrechnung zu Umsatzsteigerung infolge der Optimierung der Barrierefreiheit

Warum ist die Beachtung der Barrierefreiheit letztlich so bedeutend? Abbildung 2.10 verdeutlicht den Zusammenhang zwischen User Experience und Barrierefreiheit.

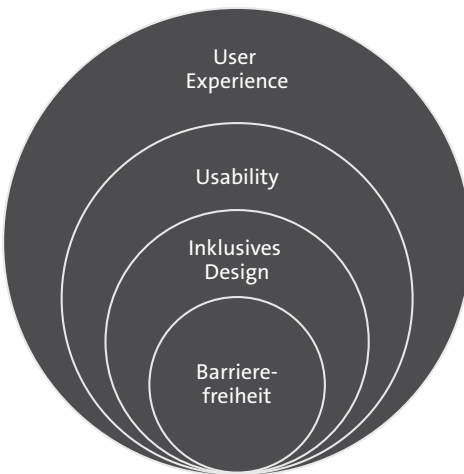


Abbildung 2.10 Zusammenhang von UX und Barrierefreiheit

Die Barrierefreiheit ist die gemeinsame Basis und Voraussetzung für die äußeren Schichten. Damit die Nutzenden ein positives Nutzungserlebnis von Ihrer Website haben, muss diese alle Menschen einbeziehen und eine hohe Usability aufweisen. Um dies zu erreichen, ist es notwendig, die Ziele, Aufgaben und Nutzungsweisen der potenziellen Nutzenden zu kennen. Um möglichst vielen Menschen eine inklusive Nutzung zu ermöglichen, d. h. unabhängig von Alter, Geschlecht, Kultur, Herkunft, Fertigkeiten und Kenntnissen etc., müssen die jeweiligen spezifischen Bedürfnisse und Eigenschaften erforscht und berücksichtigt werden. Dadurch wird die Zielgruppe maximiert.

Inklusion

Inklusion beschreibt das Bestreben, allen Menschen die gleichberechtigte und umfassende Teilhabe an allen gesellschaftlichen Prozessen zu ermöglichen – unabhängig von individuellen Fähigkeiten, Herkunft, Geschlecht oder anderen Merkmalen. Inklusion zielt auf den Abbau von Barrieren, die bestimmte Gruppen an der gleichberechtigten Teilhabe am gesellschaftlichen Leben hindern. Im Bereich der Barrierefreiheit bedeutet Inklusion insbesondere, dass Menschen mit Behinderungen nicht nur physisch, sondern auch sozial, digital und kulturell in alle Lebensbereiche integriert werden. Inklusion bedeutet, Barrieren zu erkennen und aktiv zu beseitigen. Ein inklusiver Ansatz setzt auf universelles Design, also die Gestaltung von Produkten, Dienstleistungen und Umgebungen, die von allen Menschen unabhängig von individuellen Einschränkungen genutzt werden können (weitere Details zu universellem Design siehe Abschnitt 14.6).

2.6.1 Image und Marke stärken

Tue Gutes und rede darüber – wenn sich Unternehmen dafür einsetzen, dass wirklich alle Menschen die Systeme nutzen können, schafft dies nicht nur einen größeren potenziellen Kundenkreis, sondern steigert auch das positive Image und langfristig den Markenwert. Barrierefreiheit bzw. generell Corporate Social Responsibility (CSR) kann das Image eines Unternehmens maßgeblich beeinflussen und betrifft sowohl die Innensicht als auch die Außensicht. »Wir unterstützen und fördern unsere Mitarbeitenden – egal, was im Leben passiert, wir stehen zu ihnen!« Das heißt, wie unterstützt das Unternehmen die Mitarbeitenden bei gesundheitlichen Rückschlägen? Wie barrierefrei sind die Arbeitsplätze und die technische Ausstattung? CSR umfasst häufig Initiativen zur Schaffung eines inklusiven Arbeitsumfeldes, das Menschen mit unterschiedlichen physischen und psychischen Fähigkeiten einbezieht. Dies kann neben der Bereitstellung geeigneter Technologien auch die Schulung der Mitarbeitenden zur Sensibilisierung für Barrierefreiheit umfassen. Programme, die Vielfalt und Inklusion fördern, sind ein positiver Bestandteil und können dazu beitragen,

eine Kultur der Akzeptanz und des Respekts zu schaffen, indem vielfältige Talente einbezogen und unterstützt werden. Insbesondere auf dem hart umkämpften Arbeitsmarkt sollte dies als positiver Faktor für das Unternehmen kommuniziert werden. Bildungsinitiativen können dazu beitragen, das Bewusstsein für Barrierefreiheit zu schärfen und sicherzustellen, dass alle Mitarbeitenden gleichberechtigten Zugang zu Wachstums- und Lernmöglichkeiten haben – um Mitarbeitende langfristig zu halten und zu erhalten.

Mundpropaganda (neudeutsch Word of Mouth) führt weiterhin dazu, dass Menschen sich austauschen und typischerweise Unternehmen wertschätzen, die Positives schaffen oder ermöglichen. Auch die User Experience sollte davon profitieren und gesteigert werden, wenn eine Website oder App barrierefrei ist und alle Nutzenden ein positives Erlebnis haben.

2.6.2 Mehr Power im Onlinemarketing

Suchmaschinen wie Google und Co. legen großen Wert auf qualitativ hochwertige Inhalte sowie die perfekte Übereinstimmung mit der Suchphrase Suchender. Die großen Updates von Google, die das Ranking von Webseiten in den Suchergebnissen von Google beeinflussen, wie z. B. das Panda-Update, hatten immer das Ziel, den bestmöglichen Content bereitzustellen und somit im Ranking zu priorisieren. Damit Suchmaschinen die Inhalte verstehen, werden neben den Tags und Attributen in HTML zusätzlich semantische Auszeichnungen wie Schema.org genutzt. Damit werden semantische Überschriften (H1 bis H6), Verlinkungen, Alternativtexte etc. um spezifische Informationen im Inhalt ergänzt.

Eine Studie des Anbieters [AccessibilityChecker.org](https://www.accessibilitychecker.org/guides/accessibility-seo-impact-study/) (<https://www.accessibilitychecker.org/guides/accessibility-seo-impact-study/>) zeigt, dass sich Barrierefreiheit positiv auf die Suchmaschinenoptimierung (SEO) auswirken kann. Webseiten, die die Richtlinien zur Barrierefreiheit vollständig erfüllen, haben eine höhere Chance, in den Suchergebnissen gut platziert zu werden. Die Analyse von 847 Webseiten ergab, dass nach der Implementierung von Barrierefreiheitslösungen der organische Traffic (kostenfreie Besuche über Suchmaschinen) im Durchschnitt um 12 % anstieg. Im Allgemeinen sollte eine verbesserte Barrierefreiheit zu einer besseren Benutzererfahrung führen, was wiederum die Engagement-Raten erhöht und die Verweildauer auf der Website verlängert. Darüber hinaus betont die Studie, dass Barrierefreiheitstechnologien die Auffindbarkeit der Webseiten verbessern, indem sie sie zugänglicher machen und sicherstellen, dass sie von Suchmaschinen-Crawlern leichter indexiert werden können. Für Unternehmen, die ihre SEO verbessern wollen, ist es daher ratsam, in Barrierefreiheit zu investieren. Dies kann nicht nur die Sichtbarkeit und das Ranking in Suchmaschinen verbessern, sondern auch eine inklusivere Nutzererfahrung bieten.

2.6.3 Mehr Nutzende erreichen (auch solche ohne Einschränkungen)

»Als Faustformel gilt: Barrierefreiheit im Internet ist für zehn Prozent der Bevölkerung unerlässlich, für mindestens 30 Prozent notwendig und für 100 Prozent hilfreich. Umgekehrt heißt das: wer seine Webangebote nicht barrierefrei gestaltet, schließt eine große Kundengruppe aus.«²

Dieses Zitat aus einer aktuellen Studie zeigt das große Potenzial, das in einer konsequenten Berücksichtigung der Barrierefreiheit liegt. Die Möglichkeiten der Webanalyse wurden von Unternehmen in den letzten Jahren intensiv genutzt. Tools wie Google Analytics sind auf fast allen Webseiten integriert. Wir messen, woher Nutzende kommen, welche Browser sie verwenden und welche Bildschirmauflösung sie haben. Je nach Browser reagieren wir mit responsiven Webseiten auf die Bildschirmgröße und spielen gegebenenfalls optimierte Varianten aus. Leider erfahren wir im Standard nicht, ob die Nutzenden ein assistive Technologie wie Screenreader oder Braillezeile nutzen. Daher ist das jeweilige Potenzial für das Unternehmen nur sehr schwer abzuschätzen. Es geht nicht nur um Menschen mit Behinderungen, sondern generell darum, möglichst viele Barrieren abzubauen, z. B. auch sprachliche Aspekte (Verständlichkeit, Einfachheit, Übersetzung). Aber hey, wir optimieren permanent für Suchmaschinen, warum also nicht auch für alle Menschen? Und wenn wir sie erreicht haben, können und sollten wir mit ihnen über verschiedene Kanäle kommunizieren. So ist sichergestellt, dass jeder Mensch über mindestens einen Kanal erreichbar ist.

Der demografische Wandel und die individuelle Situation zeigen, dass wir alle von barrierefreien Systemen profitieren können. Körperliche Funktionen führen im Laufe des Lebens zu »Schwachstellen«, was uns auf Hilfsmittel wie Brillen, Hörgeräte, Bewegungshilfen etc. angewiesen macht:

- ▶ Körperliche Fähigkeiten: Nachlassende Geschicklichkeit und Feinmotorik können den Umgang mit Eingabegeräten wie der Maus beeinträchtigen. Insbesondere das präzise Anklicken kleiner Interaktionselemente (Buttons, Teaser) stellt eine Herausforderung dar.
- ▶ Sehvermögen: Mit zunehmendem Alter nimmt die Fähigkeit ab, Kontraste und Farben zu erkennen und Objekte in der Nähe scharf zu sehen. Diese Veränderungen können das Lesen von Texten auf Webseiten erheblich erschweren.
- ▶ Hörvermögen: Ältere Menschen haben oft Schwierigkeiten, hohe Töne zu hören oder Audiosignale von Hintergrundgeräuschen zu unterscheiden. Das macht es schwierig, Podcasts und anderen Audioinhalten zu folgen, insbesondere wenn Hintergrundmusik vorhanden ist.

2 <https://blog.google/intl/de-de/unternehmen/engagement/barrierefreiheit-gastbeitrag-aktion-mensch/>

- Kognitive Fähigkeiten: Ein nachlassendes Kurzzeitgedächtnis, Konzentrations-schwierigkeiten und eine erhöhte Ablenkbarkeit können dazu führen, dass ältere Menschen Schwierigkeiten haben, sich auf Webseiten zurechtzufinden und Onlineaufgaben zu erledigen.

Diese altersspezifischen Einschränkungen überschneiden sich mit den Bedürfnissen von Menschen mit Behinderungen an die digitale Barrierefreiheit. Die Besonderheit liegt darin, dass gerade im Alter eine Kombination von Beeinträchtigungen auftritt: Neben dem Gehör lässt auch das Sehvermögen nach und die Motorik wird eingeschränkt. Weiterhin kommen als Folge von Krankheiten weitere Aspekte der Schwerbehinderung hinzu. Insgesamt sind 24,7 % der Ü64-Jährigen schwerbehindert (weitere Details unter https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Behinderte-Menschen/_inhalt.htm).

Rund 7 Millionen Menschen in Deutschland (Stand 2021) sind im Laufe ihres Lebens infolge einer Krankheit (z. B. Krebs) schwerbehindert geworden. Bei 258.517 Menschen war die Schwerbehinderung angeboren.³ Abbildung 2.11 zeigt die Behinderungsarten aus dem Jahr 2021 nach dem Grad der Behinderung.

Schwerbehinderte Menschen mit Ausweis (absolut und je 100.000 Einwohner). Gliederungsmerkmale: Jahre, Region, Art der Behinderung, Grad der Behinderung

Diese Tabelle bezieht sich auf:
Jahr: 2021, Region: Deutschland, Sachverhalt: Schwerbehinderte Menschen absolut

Art der Behinderung	Grad der Behinderung						
	insgesamt $\Delta \nabla$	Grad der Behinderung von 50 $\Delta \nabla$	Grad der Behinderung von 60 $\Delta \nabla$	Grad der Behinderung von 70 $\Delta \nabla$	Grad der Behinderung von 80 $\Delta \nabla$	Grad der Behinderung von 90 $\Delta \nabla$	Grad der Behinderung von 100 $\Delta \nabla$
☐ Alle Arten	7.795.340	2.677.655	1.194.030	841.845	949.255	380.195	1.752.355
Verlust oder Teilverlust von Gliedmaßen	49.895	9.920	5.610	7.350	7.940	4.485	14.590
Funktionseinschränkung von Gliedmaßen	838.860	307.370	162.545	113.015	93.295	45.440	117.195
Funktionseinschränkung der Wirbelsäule und des Rumpfes, Deformierung des Brustkorbes	795.155	371.040	150.280	95.470	69.025	32.720	76.620
Blindheit und Sehbehinderung ^{Info}	334.600	53.590	29.615	30.130	29.120	23.515	168.630
Sprach- oder Sprechstörungen, Taubheit, Schwerhörigkeit, Gleichgewichtsstörungen	318.880	87.835	48.670	44.335	41.515	20.155	76.370
Verlust einer Brust oder beider Brüste, Entstellungen u.a.	182.620	77.865	39.020	15.330	20.930	6.030	23.450
Beeinträchtigung der Funktion von inneren Organen bzw. Organsystemen	2.003.250	663.220	314.855	194.180	277.910	99.540	453.545
Querschnittslähmung, Zerebrale Störungen, Geistig-Seelische Behinderungen, Suchtkrankheiten	1.786.335	591.765	225.540	183.915	230.425	75.850	478.835
Sonstige und ungenügend bezeichnete Behinderungen	1.485.740	515.045	217.895	158.125	179.095	72.460	343.115

Abbildung 2.11 Arten der Behinderung 2021⁴

3 <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Behinderte-Menschen/Tabellen/geschlecht-behinderung.html>

4 https://www.gbe-bund.de/gbe/pkg_isgbe5.prc_menu_olap?p_uid=gast&p_aid=34510317&p_sprache=D&p_help=0&p_indnr=218&p_indsp=115&p_ityp=H&p_fid=

Daraus lässt sich schließen, dass wir alle davon betroffen sein können, insbesondere wenn wir älter werden. Damit sollten wir ein Interesse daran haben, Systeme zu schaffen, die für alle zugänglich sind.

2.7 Zusammenfassung des Kapitels

Hier sind noch einmal die wichtigsten Punkte dieses Kapitels im Überblick:

- ▶ Barrierefreiheit ist komplex und umfasst eine Vielzahl von Behinderungen, die bei der Gestaltung von Websites und Apps berücksichtigt werden müssen.
- ▶ Der menschenzentrierte Gestaltungsprozess stellt die Bedürfnisse der Nutzenden in den Mittelpunkt. Barrierefreiheit erfordert das proaktive Vermeiden von Barrieren durch Verständnis der spezifischen Bedürfnisse von Menschen mit Behinderungen.
- ▶ Neben maschinellen Tests sind User-Tests mit tatsächlichen Nutzenden, insbesondere mit Menschen mit Behinderungen, unerlässlich.
- ▶ Der kostenlose Screenreader NVDA kann zur Testung der eigenen Website aus der Perspektive von Nutzenden assistiver Technologien verwendet werden.
- ▶ Websites müssen mit Screenreadern kompatibel sein und Alternativtexte für Bilder und Videos bereitstellen. Dynamische Anpassungen wie Zoom, Kontrasterhöhung und variable Schriftgrößen sind essenziell. Strukturierte Inhalte und vollständige Tastaturbedienbarkeit erleichtern die Navigation für Sehbehinderte.
- ▶ Barrierefreiheit und ansprechendes Design schließen sich nicht aus; sie können harmonisch kombiniert werden. Die Integration von Barrierefreiheit verbessert die Usability und User Experience für alle Nutzenden.
- ▶ Herausforderungen bei der Umsetzung können durch frühzeitige Planung und bewusste Gestaltung überwunden werden.
- ▶ Kulturelle Unterschiede beeinflussen die Wahrnehmung von Farben, Symbolen und Gesten; Design sollte kultursensibel sein.
- ▶ Mehrsprachige Unterstützung und Anpassbarkeit der Benutzeroberfläche erhöhen die Barrierefreiheit. Berücksichtigung verschiedener Lesegewohnheiten (z. B. Leserichtung) und kultureller Symbolik verbessert die Zugänglichkeit für internationale Nutzende.
- ▶ Barrierefreie Websites erhöhen die Usability und reduzieren Absprünge, was zu einer Steigerung der Konversionsrate und des Deckungsbeitrags führen kann. Durch bessere Zugänglichkeit können Kundenservicekosten gesenkt und Retourenquoten reduziert werden, da die Nutzenden Informationen eigenständig finden und Produkte besser verstehen.

- ▶ Barrierefreiheit erfordert, ähnlich wie SEO und Konversionsoptimierung, laufende Anpassungen und schließt ein ansprechendes Design nicht aus.
- ▶ Unternehmen, die Systeme für alle zugänglich machen, erweitern ihre Kundenbasis und stärken ihr positives Image sowie den Markenwert langfristig.
- ▶ Initiativen zur Schaffung eines inklusiven Arbeitsumfeldes fördern Akzeptanz und Respekt, ziehen diverse Talente an und sollten als positiver Faktor im Arbeitsmarkt kommuniziert werden.
- ▶ Barrierefreie Websites erfüllen Suchmaschinenrichtlinien besser, was zu höheren Rankings und einem durchschnittlichen Anstieg des organischen Traffics führen kann.
- ▶ Semantische Auszeichnungen und barrierefreie Technologien erleichtern Suchmaschinen das Crawlen und Indexieren von Inhalten, was die Auffindbarkeit verbessert.
- ▶ Barrierefreiheit ist für 10 % der Bevölkerung unerlässlich, für 30 % notwendig und für 100 % hilfreich; nicht barrierefreie Angebote schließen große Kundengruppen aus.

Kapitel 9

Formulare – barrierefrei erstellt

*»Accessible design is good design.«
Steve Ballmer*

Die Themen im Überblick:

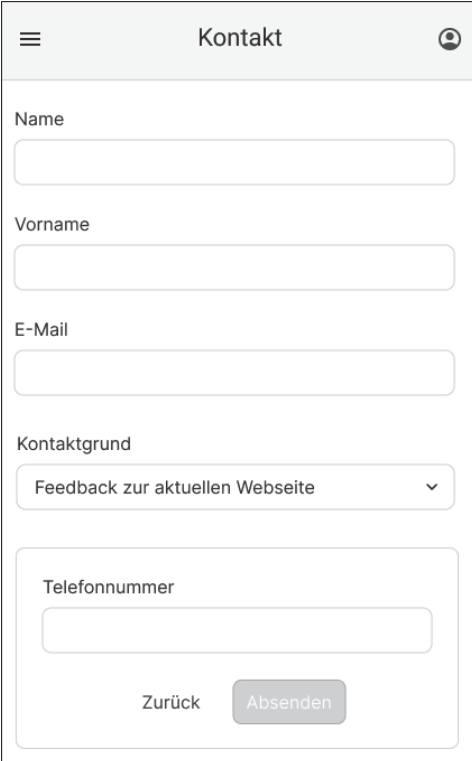
- ▶ Grundstruktur eines barrierefreien Formulars
- ▶ Eingabeelemente und mobile Tastaturlayouts in Formularen
- ▶ Beschriftungen von Formularelementen
- ▶ Unsichtbare Beschriftungen – z. B. interne Suche
- ▶ Reihenfolge und Gruppierung von Formularfeldern
- ▶ Auszeichnung von Pflichtfeldern – »required« vs. »aria-required«
- ▶ Umgang mit der Autokorrektur in Formularfeldern
- ▶ Autocomplete
- ▶ Button und Buttongestaltung (Kontrast)
- ▶ Fehlerbehandlung in Formularen
- ▶ Spezifische Eigenschaften der Eingabefelder
- ▶ Hilfestellungen
- ▶ Zusammenfassung des Kapitels

Von der Wiege bis zur Bahre – Formulare. Von der elektronischen Geburtsanzeige über die Umzugsmeldung bis hin zur Sterbeurkunde endet nahezu jeder Prozess im E-Commerce oder E-Government mit dem Ausfüllen eines Formulars. Allen gemeinsam ist, dass Daten erfasst, validiert, verarbeitet und ausgegeben werden müssen. E-Commerce-Unternehmen wollen aus den Nutzenden ihrer Website möglichst viele Kundinnen und Kunden generieren, benötigen Adressdaten und verarbeiten gegebenenfalls sensible Zahlungsinformationen. Sofern der Abschluss offline oder zu einem späteren Zeitpunkt erfolgt (Lead-Generierung), werden Kontaktdaten über ein Formular erfasst. Auch nichtkommerzielle Websites oder Blogs enthalten in der Regel interaktive Elemente wie Suchformulare oder Eingabefelder, die Informationen von den Nutzenden abfragen.

Insbesondere im E-Government (bei digitalen Behördengängen) müssen die Nutzen-
den über einen Identifikationsprozess eindeutig verifiziert werden, sollen Daten

möglichst keine Redundanzen aufweisen (Stichwort Datensparsamkeit) und sollen weiterführende Prozesse miteinander verknüpft werden (»einer für alle« lautet ein Grundsatz des E-Governments). Bei sensiblen Prozessen wie der fristgemäßen Kündigung eines Abonnements oder E-Government-Dienstleistungen wie dem Einbürgerungsantrag ist die Sorge der Nutzenden, einen Fehler zu machen, der für sie nachhaltig negative Folgen haben könnte, groß. In Usability-Tests sind immer wieder die gleichen Aussagen zu hören: »Wenn ich meine Einbürgerung elektronisch durchführe, darf ich ja keinen Fehler machen! Sonst kann das Strafen für mich bedeuten.« Die typische Behördensprache oder Warnungen vor rechtlichen Konsequenzen verstärken diese Ängste. Auf der anderen Seite sollen die Bürgerinnen und Bürger verstärkt elektronische Dienstleistungen in Anspruch nehmen können und entsprechende Angebote geschaffen werden. Da sich diese an alle richten, müssen die Formulare und Prozesse möglichst verständlich und einfach sein, was durch barrierefreie Formulare gefördert wird.

In den Bereichen SEO, Usability und Conversion wurden in den letzten Jahren weitreichende Optimierungen und Erfolge erzielt. Was würden Sie an dem Formular in Abbildung 9.1 ändern, damit es von mehr Menschen zufriedenstellend genutzt werden kann?



The image shows a mobile contact form titled "Kontakt". At the top, there is a hamburger menu icon on the left, the title "Kontakt" in the center, and a profile icon on the right. Below the header, there are five input fields: "Name", "Vorname", "E-Mail", "Kontaktgrund" (a dropdown menu with "Feedback zur aktuellen Webseite" selected), and "Telefonnummer". At the bottom, there are two buttons: "Zurück" (disabled) and "Absenden" (active).

Abbildung 9.1 Beispielhaftes Kontaktformular mit Optimierungspotenzial

Interessant wird es, wenn die Sicht sowohl der Usability als auch der Barrierefreiheit einfließt.

Abbildung 9.2 zeigt potenzielle Fragen von Nutzenden aus Sicht der Usability und aus Sicht der Barrierefreiheit, die sich optimal ergänzen.

The image shows a contact form titled 'Kontakt' with several input fields and a submit button. Annotations point to specific areas of the form:

- Name:** An annotation asks 'Pflichtfeld? Optional?' (Required field? Optional?).
- Vorname:** An annotation notes 'Kontrast 1,3:1 (zu gering)' (Contrast 1.3:1 (too low)).
- E-Mail:** No annotation is present.
- Kontaktgrund:** A dropdown menu with the selected option 'Feedback zur aktuellen Webseite'. An annotation asks 'Darstellung bei Zoom?' (Display when zoomed?).
- Telefonnummer:** An annotation asks 'Format? +49 ?' (Format? +49?).
- Zurück / Absenden:** An annotation notes 'Kontrast 1,4:1 (zu gering, wenn Button aktiv)' (Contrast 1.4:1 (too low, when button is active)).

Abbildung 9.2 Optimierungen des Formulars aus Sicht der Usability und Barrierefreiheit


Folgende Optimierungen ergeben sich daraus:

- ▶ Der Kontrast des Rahmens bei den Eingabefeldern ist zu gering. Dies führt für Menschen mit einer Sehbehinderung zu Problemen.
- ▶ Im Formular sind Pflichtfelder nicht entsprechend hervorgehoben. Nutzende werden sich die Frage stellen, welche von den Feldern nun Pflichtfelder sind und welche nicht. Screenreader übermitteln die Information eines Pflichtfeldes durch das Attribut `required`.
- ▶ Sofern die Nutzenden einen Zoom aufgrund einer Sehbehinderung eingestellt haben, wird der Inhalt des Feldes »Kontaktgrund« nicht mehr lesbar sein, da der Inhalt »Feedback zur aktuellen Webseite« die vorgegebene Breite des Feldes überschreiten wird.
- ▶ Ein Eingabeformat der Telefonnummer sollte vorgegeben oder die Eingabe so unterstützt werden, dass keine Fehleingabe möglich ist. So könnten z. B. das Leerzeichen oder der Schrägstrich im Feld zugelassen werden.

- ▶ Fehlermeldungen durch Fehleingaben oder nicht ausgefüllte Formularfelder müssen so dargestellt werden, dass sie sowohl für Menschen mit Sehbehinderung direkt nach Abschicken des Formulars im Fokus sind als auch für Screenreader-Nutzende ebenfalls nach dem Abschicken des Formulars unmittelbar ausgegeben werden.
- ▶ Des Weiteren müssen Fehlermeldungen für alle Menschen verständlich formuliert sein. Eine Meldung »Bitte füllen Sie dieses Feld aus!« ist dafür nicht ausreichend.

9.1 Grundstruktur eines barrierefreien Formulars

Ein Formular ermöglicht die Eingabe von Daten, die anschließend in der Regel an einen Server zur weiteren Verarbeitung gesendet werden. Es besteht aus verschiedenen Elementen wie Textfeldern, Auswahlménüs, Buttons und anderen Eingabefeldern, die es ermöglichen, Informationen an die Website-Betreiber zu übermitteln. Abbildung 9.3 zeigt ein einfaches Registrierungsformular.



Das Bild zeigt ein einfaches Registrierungsformular in einem rechteckigen Rahmen. Oben links steht der Titel 'Registrierung'. Darunter befinden sich vier Textfelder, jeweils mit einer Beschriftung links daneben: 'Name:', 'E-Mail:', 'Passwort:' und 'Passwort bestätigen:'. Jedes Textfeld ist ein leeres, rechteckiges Eingabefeld. Unter dem letzten Feld befindet sich ein Button mit der Aufschrift 'Registrieren'.

Abbildung 9.3 Einfaches Registrierungsformular

Das Formular hat in HTML die folgende Grundstruktur:

1. `<form>`-Element: Dieses Element umschließt alle Komponenten des Formulars. Attribute wie `action` (die URL, an die die Formulardaten gesendet werden) und `method` (die Methode zum Senden der Daten, üblicherweise GET oder POST) definieren, wie und wohin die Formulardaten übermittelt werden.
2. Eingabefelder (`<input>`, `<textarea>`, `<select>` etc.): Diese Elemente ermöglichen den Nutzenden die Eingabe von Daten. Der Typ des `<input>`-Elements (z. B. `text`, `email`, `password`) bestimmt das Aussehen und welche Art von Daten eingegeben werden können.

3. Beschriftungen (<label>): Für jedes Eingabefeld sollte es eine Beschriftung geben, die deutlich macht, welche Information in diesem Feld erwartet wird. Das `for`-Attribut des <label>-Elements muss mit der `id` des zugehörigen Eingabefeldes übereinstimmen, um eine eindeutige Verbindung zwischen beiden Elementen zu schaffen und so die Zugänglichkeit zu verbessern.
4. Absende-Button (<input type="submit"> oder <button type="submit">): Dieses Element sendet die in das Formular eingegebenen Daten an den Server.

Listing 9.1 zeigt den Code für die grundsätzliche Struktur des HTML-Formulars. Der <input type="submit">-Button am Ende des Formulars ermöglicht das Absenden des Formulars. Das <fieldset> mit seinem <legend>-Element gruppiert zusammengehörige Bereiche und gibt dem Formular in Abbildung 9.3 eine Struktur, die sowohl die visuelle Darstellung als auch die Zugänglichkeit verbessert.

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Registrierungsformular</title>
  <style>
    form {
      max-width: 400px;
      margin: 0 auto;
    }
    label, input {
      display: block;
      width: 100%;
      margin-bottom: 1em;
    }
    input[type="submit"] {
      width: auto;
      margin-top: 1em;
    }
  </style>
</head>
<body>
<form action="/registrierung" method="post">
  <fieldset>
    <legend>Registrierung</legend>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required>
    <label for="email">E-Mail:</label>
    <input type="email" id="email" name="email" required>
```

```

<label for="passwort">Passwort:</label>
<input type="password" id="passwort" name="passwort" required>
<label for="passwortBestaetigen">Passwort bestätigen:</label>
<input type="password" id="passwortBestaetigen" name=
"passwortBestaetigen" required>
<input type="submit" value="Registrieren">
</fieldset>
</form>
</body>
</html>

```

Listing 9.1 Registrierungsformular mit grafischen Anpassungen

In den folgenden Abschnitten werden die Struktur und die Elemente erläutert, die zur Erstellung eines benutzerfreundlichen und zugänglichen Formulars verwendet werden.

9.2 Eingabelemente und mobile Tastaturlayouts in Formularen

In Formularen haben Sie verschiedene Möglichkeiten, die gewünschten Daten ihrer Nutzenden in einem spezifischen Format abzufragen. Das `<input>`-Element ist eines der vielseitigsten Elemente in HTML-Formularen. Es unterstützt verschiedene `type`-Attribute, die definieren, welche Art von Daten eingegeben werden können. Abbildung 9.4 zeigt ein Formular mit verschiedenen Eingabelementen, wie z. B. Radiobutton, Checkbox oder Passwortfeld.

Vorname: Passwort: E-Mail: Alter:

Geschlecht

Weiblich Männlich Divers

Lebenslauf hochladen: Keine ausgewählt Newsletter abonnieren

Abbildung 9.4 Formular mit unterschiedlichen Eingabelementen

Tabelle 9.1 zeigt die gängigsten `<input>`-Typen für ein Formular.

Input-Typ	Zweck	Beispiel
text	Erlaubt die Eingabe einer einzelnen Zeile Text.	<code><input type="text" id="vorname" name="vorname"></code>
password	ein Textfeld, das die Eingabe verdeckt (Anzeige *), geeignet für Passwörter	<code><input type="password" id="password" name="password"></code>

Tabelle 9.1 Grundlegende `<input>`-Typen

Input-Typ	Zweck	Beispiel
email	ein Textfeld mit Validierung der Struktur für E-Mail-Adressen. Prüft nicht, ob es die E-Mail tatsächlich gibt.	<code><input type="email" id="email" name="email"></code>
checkbox	ein Kästchen, das markiert werden kann, z. B. für Zustimmungen	<code><input type="checkbox" id="agree" name="Newsletter abonnieren"></code>
number	ein Eingabefeld für numerische Werte mit optionalen Attributen wie min, max und step	<code><input type="number" id="alter" name="alter" min="18" step="1"></code>
date	ein Eingabefeld zur Auswahl eines Datums	<code><input type="date" id="date" name="date"></code>
radio	Radiobuttons, aus denen die Nutzenden eine Option auswählen können	<code><input type="radio" id="gender_female" name="gender" value="Weiblich"></code> <code><input type="radio" id="gender_male" name="gender" value="Männlich"></code> ...
file	ein Button, um eine oder mehrere Dateien hochzuladen	<code><input type="file" id="lebenslauf" name="lebenslauf" accept=".pdf"></code>
submit/ reset/ button	ein Button, um das Formular zu übermitteln bzw. abubrechen	<code><input type="submit" value="Anmelden"></code>

Tabelle 9.1 Grundlegende `<input>`-Typen (Forts.)

Ein großes Usability-Problem ist die richtige Tastatur im mobilen Umfeld. Wenn die Nutzenden auf ihrem Smartphone pro Formularfeld zur »richtigen« Tastatur wechseln müssen, ist das nicht nur nervig, sondern auch eine Barriere. Bei der Eingabe einer E-Mail-Adresse sollte z. B. eine Tastatur angezeigt werden, die das @-Zeichen enthält, wie in Abbildung 9.5 dargestellt.

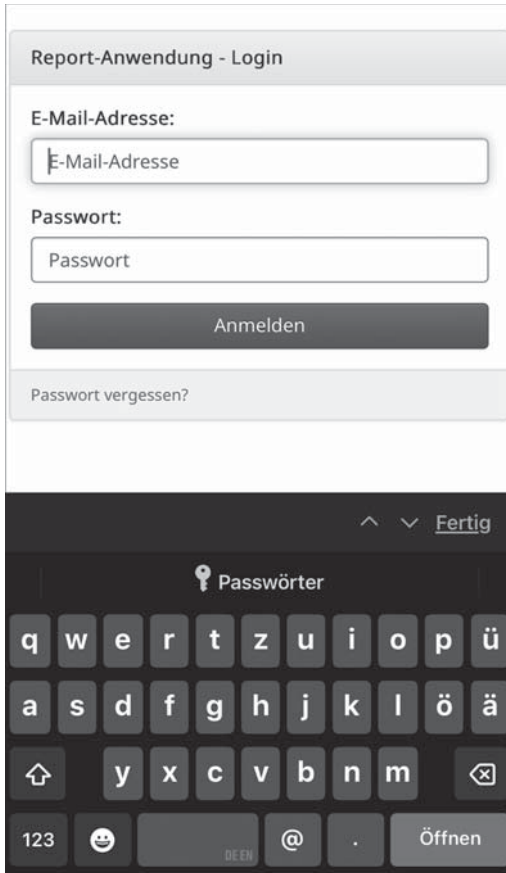


Abbildung 9.5 Richtige Tastaturauswahl je nach Eingabefeld – hier die E-Mail-Adresse

Für die Eingabe von Zahlen sollte eine numerische Tastatur eingeblendet werden. So können die Nutzenden schnell und einfach durch das Formular geführt werden. Das `<input>`-Element in HTML-Formularen beeinflusst das Layout der Tastatur auf Smartphones und Tablets. Dadurch wird die Dateneingabe erleichtert und die Benutzererfahrung verbessert. Der Browser kann dann ein angepasstes Tastaturlayout anzeigen, das für die Eingabe dieser spezifischen Art von Daten optimiert ist. Tabelle 9.2 zeigt, wie verschiedene `type`-Werte das Tastaturlayout beeinflussen.

Type-Attribut	Auswirkung
<code>type="text"</code>	Zeigt eine Standardtastatur für die Texteingabe an. Dies ist das grundlegendste Layout, geeignet für allgemeine Texteingaben.

Tabelle 9.2 Das Attribut »type« im Formularfeld und seine Auswirkung auf das Tastaturlayout


Type-Attribut	Auswirkung
type="tel"	Zeigt eine Tastatur mit einem Zahlenlayout an, das für die Eingabe von Telefonnummern optimiert ist. Diese Tastatur enthält Ziffern und Symbole wie Klammern und das Pluszeichen für internationale Telefonnummern.
type="number"	Aktiviert eine Tastatur, die hauptsächlich Zahlen und eine begrenzte Anzahl von Symbolen wie das Dezimaltrennzeichen anzeigt. Diese Tastatur ist ideal für die Eingabe numerischer Werte.
type="email"	Diese Tastatur enthält zusätzlich zu den Buchstaben und Zahlen spezielle Tasten für Symbole, die häufig in E-Mail-Adressen verwendet werden, wie z. B. @ und ».«. Dies erleichtert die Eingabe von E-Mail-Adressen.
type="url"	Zeigt eine Tastatur mit speziellen Tasten für die Eingabe von Webadressen, einschließlich einer ».com«-Taste und anderer nützlicher Symbole wie »/«, um die Eingabe von URLs zu vereinfachen.
type="search"	Zeigt eine Tastatur mit einer dedizierten »Suchen«-Taste, die das Absenden einer Suchanfrage vereinfacht, anstatt die  -Taste verwenden zu müssen.
type="date", type="time", type="datetime- local"	Anstelle einer Tastatur kann eine Datums- oder Zeitauswahl angezeigt werden, um die Eingabe von Datum und/oder Uhrzeit zu erleichtern. Diese Bedienelemente sind speziell für die Auswahl von Datum und Uhrzeit optimiert und verwenden Scrollräder oder Kalender-Widgets

Tabelle 9.2 Das Attribut »type« im Formularfeld und seine Auswirkung auf das Tastaturlayout (Forts.)

Die Anpassung des Tastaturlayouts auf der Grundlage des Eingabetyps verbessert nicht nur die Benutzerfreundlichkeit, sondern trägt auch zur Verringerung von Eingabefehlern bei. Wählen Sie daher das type-Attribut sorgfältig aus, um sicherzustellen, dass Ihre Nutzenden die bestmögliche Tastatur für die jeweilige Eingabe erhalten.

Tipp: Tastaturprüfung auf dem Smartphone

Prüfen Sie auf Ihrem Smartphone oder Tablet, welche Tastatur für das jeweilige Formularfeld angezeigt wird, und passen Sie diese unbedingt an – in einer mobilen Welt sollte dies der Standard sein.

9.3 Beschriftungen von Formularelementen

Die Beschriftung von Steuerelementen in HTML-Formularen ist ein wesentlicher Aspekt der Zugänglichkeit und Gebrauchstauglichkeit von Webseiten. Die Richtlinie WCAG 3.3.2, »Beschriftungen (Labels) oder Anweisungen«, besagt, dass Labels oder Anweisungen zur Verfügung gestellt werden müssen, wenn der Inhalt Eingaben von Nutzenden erfordert. Dies hilft allen Nutzenden zu verstehen, welche Eingaben in Formularfeldern erwartet werden. Labels sind nicht nur für die Barrierefreiheit wichtig, sondern verbessern auch die allgemeine Benutzererfahrung, indem sie die Lesbarkeit und Verständlichkeit von Formularen erhöhen. Abbildung 9.6 zeigt ein einfaches Beispiel für ein Formular, das die Anforderungen von WCAG 3.3.2 erfüllt.



Abbildung 9.6 Feld mit verpflichtender Eingabe

Gegebenenfalls ist Ihnen beim Betrachten der Abbildung aufgefallen, dass die Anordnung und die Fehlermeldung nicht gut sind, dazu mehr in Abschnitt 9.10, »Fehlerbehandlung in Formularen«. Jedes Formulareingabeelement hat ein zugehöriges `<label>`, das durch das `for`-Attribut mit der `id` des Eingabeelements verknüpft ist (siehe Listing 9.2).

```
<form>
  <label for="username">Benutzername:</label>
  <input type="text" id="username" name="username" required>
  <label for="password">Passwort:</label>
  <input type="password" id="password" name="password" required>
  <button type="submit">Einloggen</button>
</form>
```

Listing 9.2 Beschriftung von Formularfeldern (Label) mittels `for`-Attribut

Die `for`-Verbindung ermöglicht Screenreadern die korrekte Zuordnung des Labels zum Eingabefeld. Dies erleichtert es den Nutzenden zu verstehen, was in jedes Feld eingegeben werden muss. Die Labels »Benutzername:« und »Passwort:« dienen als klare Anweisungen, was in die Felder eingegeben werden muss. Das `required`-Attribut signalisiert auch den assistiven Technologien, dass das Feld erforderlich ist. Es ergänzt die visuelle Anzeige der Erforderlichkeit (z. B. durch ein Sternchen*) in einer für Screenreader lesbaren Form. Gleichzeitig wird ein Standardfehler vom Browser und den assistiven Technologien ausgegeben, wenn das Feld nicht ausgefüllt ist.

Zusammenfassend lassen sich folgende Empfehlungen für die Beschriftung von Steuerelementen geben:

1. Klare Bezeichnungen: Verwenden Sie präzise und klare Bezeichnungen, die genau beschreiben, welche Information oder Eingabe erwartet wird.
2. Verknüpfung von Label und Eingabeelement: Stellen Sie sicher, dass jedes `<input>`-Element mit einem `<label>` verknüpft ist. Verwenden Sie das `for`-Attribut des `<label>`-Elements, um es mit der `id` des entsprechenden `<input>`-Elements zu verbinden.
3. Direkte Nähe: Platzieren Sie Labels in unmittelbarer Nähe zu ihren Eingabeelementen, um die visuelle Zuordnung für alle Nutzenden zu erleichtern (»Gesetz der Nähe« in der Gestaltung), insbesondere für Menschen mit kognitiven oder visuellen Einschränkungen.
4. Zusätzliche Anweisungen: Wenn spezielle Formate oder Einschränkungen für die Eingabe gelten (z. B. ein bestimmtes Datumsformat wie TT.MM.JJJJ oder Passwortregeln), geben Sie klare Anweisungen, entweder direkt bei dem Label oder in der Nähe des Eingabeelements. Eingabehinweise müssen programmatisch mit dem Eingabefeld assoziiert werden, z. B. als (Teil des) `<label>`, per `aria-describedby`.

Testen Sie die Zugänglichkeit Ihrer Formulare mit verschiedenen assistiven Technologien (mindestens Screenreader) und Benutzereingabemethoden (mindestens Spracheingabe und ausschließliche Tastaturnutzung), um sicherzustellen, dass alle Nutzenden sie effektiv ausfüllen können.

9.4 Unsichtbare Beschriftungen – z. B. interne Suche

Um ein internes Suchfeld so zu beschriften, dass es für assistive Technologien gut zugänglich ist, sollte sichergestellt werden, dass das Suchfeld ein eindeutig identifizierbares Label hat. Wenn die Beschriftung aus Gründen der intuitiven Bedienbarkeit nicht sichtbar sein soll (z. B. bei Suchfeldern), kann dies über CSS geändert werden. Für Screenreader sollten jedoch ARIA-Attribute (`aria-label`, `aria-labelledby`, `aria-describedby`) verwendet werden, um die Zugänglichkeit zu verbessern. Listing 9.3 zeigt eine Auszeichnung mittels ARIA.

```
<!-- Wenn ein sichtbares Label nicht geeignet ist -->
<input type="search" id="site-search" name="q" aria-label="Website
durchsuchen">
<!-- Verwendung von aria-labelledby und aria-describedby für zusätzlichen
Kontext -->
<label id="searchLabel" class="visually-hidden">Suche</label>
```

```
<input type="search" aria-labelledby="searchLabel" aria-describedby="searchTip" name="q">
<div id="searchTip">Geben Sie Stichworte ein, um die Website zu durchsuchen</div>
```

Listing 9.3 Auszeichnung von Feldern mit ARIA


Die Verwendung von ARIA-Attributen kann zusätzliche kontextuelle oder beschreibende Informationen liefern, insbesondere wenn eine visuelle Beschriftung das Design beeinträchtigt oder schwierig zu implementieren ist (wie in Abschnitt 4.4, »WAI ARIA«, erläutert):

- ▶ `aria-label` bietet eine unsichtbare Beschriftung, die nur von assistiven Technologien (Screenreadern) gelesen werden kann. Sie ist nützlich, wenn keine sichtbare Beschriftung vorhanden ist (d. h., wenn aus dem optischen Kontext eindeutig erkennbar ist, was in das Feld eingegeben wird, wie z. B. die PLZ für die Suche nach der nächstgelegenen Serviceniederlassung).
- ▶ `aria-labelledby` verweist auf die ID eines anderen Elements, das als Label verwendet wird. Es sind auch mehrere IDs zulässig, jeweils getrennt durch Leerzeichen.
- ▶ `aria-describedby` kann verwendet werden, um auf ein Element zu verweisen, das eine Beschreibung des Suchfeldes enthält, falls zusätzliche Anweisungen erforderlich sind. Auch hier sind mehrere IDs zulässig, jeweils getrennt durch Leerzeichen.


Für die Zugänglichkeit eines Suchfeldes ist es entscheidend, dass es korrekt beschriftet ist und von assistiven Technologien leicht erkannt werden kann. Die Kombination aus visueller Beschriftung (`<label>`), gegebenenfalls klarem `placeholder`-Text und ARIA-Attributen (`aria-label`, `aria-labelledby`, `aria-describedby`) stellt sicher, dass alle Nutzenden das Suchfeld effektiv nutzen können.

9.5 Reihenfolge und Gruppierung von Formularfeldern

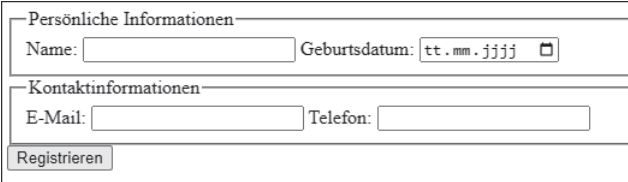
Die Reihenfolge, in der Formularfelder in Formularen angeordnet sind, spielt eine wichtige Rolle für die Benutzerfreundlichkeit und Zugänglichkeit. Sie bestimmt, wie die Nutzenden mit und ohne assistive Technologien durch ein Formular navigieren. Eine logische und intuitive Anordnung der Felder kann die Formulareingabe beschleunigen, Missverständnisse reduzieren und die Usability verbessern.

Wie in Abschnitt 3.4, »DOM und Accessibility Tree«, erläutert, ergibt sich die Reihenfolge der Formularfelder aus ihrer Position im HTML-Code, genauer gesagt im DOM. Browser und assistive Technologien folgen dieser Reihenfolge, wenn sie den Inhalt eines Formulars anzeigen oder wenn die Nutzenden die -Taste verwenden, um

von einem Feld zum nächsten zu springen. Die folgenden Best Practices sollten Sie daher berücksichtigen:

- ▶ **Natürlicher Fluss:** Ordnen Sie die Formularfelder in einer Reihenfolge an, die einem natürlichen Konversationsfluss entspricht. Zum Beispiel sollten persönliche Informationen wie Name und Adresse vor Kontaktinformationen wie E-Mail und Telefonnummer stehen.
- ▶ **Sequenzielle Navigation:** Achten Sie darauf, dass die Tab-Reihenfolge der logischen Anordnung der Formularelemente entspricht. In der Regel bedeutet dies, dass die Reihenfolge der Elemente im HTML-Code die Reihenfolge bestimmt, in der die Nutzenden mit der -Taste durch die Felder navigieren.
- ▶ **Gruppieren Sie zusammengehörige Informationen:** Verwenden Sie `<fieldset>` und `<legend>`, um verwandte Elemente zu gruppieren und zu bezeichnen (z. B. Lieferinformationen im Check-out-Prozess). Dies hilft den Nutzenden, die Informationen im Kontext zu verstehen.

Die Gruppierung von Formularinhalten in HTML ist eine effektive Methode, um zusammengehörige Informationen in einem Formular zu organisieren und die Benutzerführung zu verbessern. Abbildung 9.7 zeigt die visuelle Trennung von persönlichen Informationen und Kontaktinformationen.



Persönliche Informationen	
Name: <input type="text"/>	Geburtsdatum: <input type="text" value="tt.mm.jjjj"/>
Kontaktinformationen	
E-Mail: <input type="text"/>	Telefon: <input type="text"/>
<input type="button" value="Registrieren"/>	

Abbildung 9.7 Gruppierung von Inhalten

Die Hauptelemente für die Gruppierung von Formularinhalten sind `<fieldset>` und `<legend>`. Das Element `<fieldset>` in Listing 9.4 wird verwendet, um eine Gruppe von Formularelementen zu bilden. Es hilft nicht nur visuell durch die Gruppierung von Elementen, sondern auch Screenreadern, die Zusammengehörigkeit von Elementen zu erkennen. Das `<legend>`-Element gibt eine Überschrift für die `<fieldset>`-Gruppe an. Es bietet Kontext und erleichtert damit das Verständnis der Gruppe von Formularelementen für alle Nutzenden.

```
<form>
  <fieldset>
    <legend>Persönliche Informationen</legend>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required>
```

```

    <label for="birthdate">Geburtsdatum:</label>
    <input type="date" id="birthdate" name="birthdate" required>
</fieldset>

<fieldset>
  <legend>Kontaktinformationen</legend>
  <label for="email">E-Mail:</label>
  <input type="email" id="email" name="email" required>

  <label for="phone">Telefon:</label>
  <input type="tel" id="phone" name="phone">
</fieldset>

  <button type="submit">Registrieren</button>
</form>

```

Listing 9.4 Gruppierung von Inhalten mittels »fieldset« und »legend«

Eine Gruppenbeschriftung ist zwingend erforderlich für Gruppen von Auswahlaltern (Radiobuttons) und gegebenenfalls sinnvoll bei Gruppen von Kontrollkästchen (Checkboxes).

9.6 Auszeichnung von Pflichtfeldern – »required« vs. »aria-required«

Bei Formularen ist die Frage spannend, welche Elemente obligatorisch und welche optional (freiwillig) sind. Hier gibt es unterschiedliche Ansätze und Meinungen in den Unternehmen. Ist die Angabe der Telefonnummer bei einer Bestellung zwingend notwendig oder optional? Dient die Telefonnummer der Bonitätsprüfung und wird deshalb abgefragt? Wie verhält es sich, wenn die Nutzenden die Zahlart PayPal wählen, dann wird sie im Prinzip gar nicht mehr benötigt – dennoch sammeln?

Wenn ein `<input>`-Element in HTML mit dem Attribut `required` versehen ist, wird das optionale Eingabefeld zu einem Pflichtfeld. Das Formular wird nicht abgeschickt, wenn dieses Feld nicht ausgefüllt wurde. Der Browser zeigt normalerweise eine Meldung an, die den Nutzenden auffordert, das Pflichtfeld auszufüllen. Da in der Regel auf dem Server nochmals eine Validierung stattfindet, verzichten viele Websites auf die Validierung direkt im Browser. Der Unterschied zwischen `required` und `aria-required` in HTML-Formularen liegt hauptsächlich in ihrer Funktion und Anwendung. Das Attribut `aria-required` signalisiert assistiven Technologien, dass ein Feld erforderlich ist. Im Gegensatz zum Attribut `required`, das eine eingebaute Validierungsfunktion in Browsern auslöst, dient `aria-required` ausschließlich dazu, die Zu-

gänglichkeitsinformationen zu verbessern, ohne automatische Validierung oder Fehlermeldungen bereitzustellen. `aria-required="true"` teilt assistiven Technologien explizit mit, dass ein Feld ausgefüllt werden muss. Dies ist besonders in Situationen nützlich, in denen die Verwendung des `required`-Attributs nicht möglich oder nicht erwünscht ist, z. B. bei nicht standardmäßigen Formularelementen, die mit JavaScript erstellt wurden. In der Praxis wird empfohlen, das `required`-Attribut zu verwenden, um von der nativen Browserunterstützung zu profitieren, und `aria-required` nur in spezifischen Anwendungsfällen zu nutzen, um die Zugänglichkeit zu erhöhen.

9.7 Umgang mit der Autokorrektur in Formularfeldern

Bei E-Mail-Adressen müssen zwangsläufig Namen verwendet werden, die im Idealfall nicht durch die Autokorrektur des mobilen Browsers verändert werden sollten. (tobaub wird nach Eingabe zu Autobahn korrigiert, obwohl *tobaub@domain.de* eine valide E-Mail-Adresse sein kann. So etwas ist ärgerlich.) Zur Steuerung steht in Formularen das standardisierte HTML5-Attribut `spellcheck` zur Verfügung, das angibt, ob der Inhalt eines Elements auf Rechtschreibfehler überprüft werden soll oder nicht. Im Gegensatz zu `autocorrect` in iOS (Safari) führt `spellcheck` keine automatische Korrektur durch, sondern hebt potenzielle Rechtschreibfehler hervor, um die Aufmerksamkeit der Nutzenden auf diese zu lenken. Das Attribut `spellcheck` kann auf `spellcheck="true"` oder `spellcheck="false"` gesetzt werden, um die Rechtschreibprüfung für das Element zu aktivieren bzw. zu deaktivieren. Die tatsächliche Darstellung und Funktionsweise kann je nach Browser und Betriebssystem variieren. Die Verwendung von `autocorrect` auf iOS-Geräten reduziert Tippfehler, indem Wörter, die auf der Grundlage des Wörterbuches des Geräts oder der aktuellen Spracheinstellung als falsch geschrieben erkannt werden, automatisch korrigiert werden.

In Listing 9.5 ist die automatische Korrektur für das `<textarea>`-Element deaktiviert (`autocorrect="off"`), während die Rechtschreibprüfung aktiviert ist (`spellcheck="true"`).

```
<textarea autocorrect="off" spellcheck="true">Hier können Sie Ihren Text eingeben...</textarea>
```

Listing 9.5 Autokorrektur und Rechtschreibprüfung im Formular

Dies könnte in einem Kontext nützlich sein, in dem die Nutzenden genauere Kontrolle über ihre Texteingabe haben möchten, ohne durch automatische Korrekturen gestört zu werden, aber dennoch von Hinweisen auf mögliche Rechtschreibfehler profitieren möchten.

9.8 Autocomplete

Angemessene sichtbare Beschriftungen und Anweisungen können den Nutzenden helfen, den Zweck von Eingabefeldern in Formularen zu verstehen. Alle Nutzenden können jedoch von semantisch eindeutigen und sprachunabhängigen Felddefinitionen profitieren, sodass Felder automatisch ausgefüllt werden können, wenn dies gewünscht wird. Das heißt, wenn sich Felder auf persönliche Daten (Name, Telefonnummer etc.) beziehen, können Browsertools wie *Autofill* verwendet werden, um

- ▶ die Felder automatisch auszufüllen (autocomplete) oder
- ▶ zusätzliche Informationen wie Bilder über spezielle Plug-ins anzuzeigen.

Für einige Eingabefelder bietet das Attribut `type` bereits eine Möglichkeit, die Art der erwarteten Eingabe grob anzugeben, z. B. `<input type="email">`. Dies sind jedoch nur sehr grobe Kategorien, die den Typ der Eingabe beschreiben, aber nicht unbedingt den Zweck, insbesondere bei benutzerspezifischen Eingabefeldern. Das `autocomplete`-Attribut akzeptiert nur bestimmte Werte. Dies ermöglicht eine feinere Definition oder Identifizierung des Zwecks als es das `type`-Attribut vermag, z. B. durch die Möglichkeit, einen spezifischen Typ von Namen anzugeben:

- ▶ Nachname (`autocomplete="name"`),
- ▶ Vorname (`autocomplete="given-name"`),
- ▶ Familienname (`autocomplete="family-name"`),
- ▶ Benutzername (`autocomplete="username"`)

WCAG 1.3.5, »Identifizierung des Eingabezwecks« (Level AA), zielt darauf ab, die Zugänglichkeit und zugleich die Benutzerfreundlichkeit von Webformularen zu verbessern, indem gefordert wird, dass der Zweck von Eingabefeldern programmatisch bestimmbar sein muss. Dies ermöglicht assistiven Technologien und automatisierten Hilfsmitteln, Nutzenden bei der Eingabe von Informationen zu unterstützen, indem sie beispielsweise vorausgefüllte Daten anbieten oder spezifische Eingabeanforderungen verstehen. Abbildung 9.8 zeigt die Möglichkeit des Ausfüllens mit dem individuellen Vornamen.

The image shows a web form with two input fields. The top field is labeled 'Vorname*' and contains the placeholder text 'Vornamen eingeben ...'. Below it, a dropdown menu is open, showing a single suggestion: 'Tobias'. The bottom field is labeled 'Neue E-Mail-Adresse*' and contains the placeholder text 'Neue E-Mail-Adresse eingeben'.

Abbildung 9.8 Autocomplete mit dem Vornamen

Listing 9.6 zeigt den Code für ein Registrierungsformular für neue Nutzende. Um WCAG 1.3.5 zu entsprechen, verwendet die Website Formularfeldattribute, um den Zweck jedes Eingabefeldes anzugeben.

```

<form>
  <label for="first-name">Vorname:</label>
  <input type="text" id="first-name" name="given-name" autocomplete="given-
name">

  <label for="email">E-Mail:</label>
  <input type="email" id="email" name="email" autocomplete="email">

  <label for="new-password">Passwort erstellen:</label>
  <input type="password" id="new-password" name="new-password" autocomplete=
"new-password">

  <button type="submit">Registrieren</button>
</form>

```

Listing 9.6 Integration von »autocomplete« zur Optimierung der Eingabe

In diesem Beispiel verwendet jedes `<input>`-Element das `autocomplete`-Attribut mit einem spezifischen Wert, der den Zweck des Feldes identifiziert (z. B. `given-name` für den Vornamen, `email` für die E-Mail-Adresse und `new-password` für das neue Passwort). Diese Kennzeichnung ermöglicht es assistiven Technologien und Browsern, die Felder korrekt zu identifizieren und gegebenenfalls automatisch auszufüllen.

9.9 Button und Buttongestaltung (Kontrast)

Am Ende des Formulars befinden sich typischerweise ein oder mehrere Buttons (Schaltflächen), um das Formular abzuschicken oder zu verlassen. Das `<button>`-Element kann zum Absenden von Formularen, zum Zurücksetzen von Formulardaten oder für allgemeine Aktionen verwendet werden, die über JavaScript gesteuert werden. Mit `<input type="button">`, `<input type="submit">` oder `<input type="reset">` bietet das `<button>`-Element umfassende gestalterische Freiheiten, da es nicht nur Text, sondern auch HTML-Inhalte wie Bilder oder Formatierungen enthalten kann. Abbildung 9.9 zeigt ein HTML-Formular mit Buttons zum Absenden (`submit`), Zurücksetzen (`reset`) oder klassischen Anklicken (`button`).

Das Bild zeigt ein HTML-Formular mit zwei Textfeldern. Das obere Feld ist mit 'Name:' beschriftet, das untere mit 'E-Mail:'. Darunter befinden sich drei Buttons nebeneinander: 'Formular absenden', 'Formular zurücksetzen' und 'Klick mich'.

Abbildung 9.9 Beispiel für Buttons in Formularen

Listing 9.7 zeigt die unterschiedliche Konfiguration der jeweiligen Aktionen der Buttons aus Abbildung 9.9.

```
<form action="/submit-form" method="post">
  <div>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
  </div>
  <div>
    <label for="email">E-Mail:</label>
    <input type="email" id="email" name="email">
  </div>
  <!-- Submit Button -->
  <button type="submit">Formular absenden</button>
  <!-- Reset Button -->
  <button type="reset">Formular zurücksetzen</button>
  <!-- Allgemeiner Button -->
  <button type="button" onclick="alert('Hallo Welt!')">Klick mich</button>
</form>
```

Listing 9.7 Buttons in Formularen und deren Definition

Der Submit-button (`<button type="submit">`) wird verwendet, um das Formular zu übermitteln. Zum Abbrechen kann der Reset-Button (`<button type="reset">`) verwendet werden, der alle Eingabefelder des Formulars auf ihre ursprünglichen Werte zurücksetzt. Dies ist nützlich, wenn die Nutzenden von vorn beginnen möchten. Der Standardbutton (`<button type="button">`) dient als allgemeiner Button für verschiedene Zwecke. In diesem Beispiel wird JavaScript (`onclick`-Attribut) verwendet, um eine einfache Aktion auszulösen, wenn der Button angeklickt wird (hier Anzeigen einer Mitteilung in Form einer Alert-Box). Das `<button>`-Element kann mit CSS gestaltet werden, um das Aussehen der Buttons individuell anzupassen. Achten Sie darauf, dass alle Buttons eine klare Beschreibung ihres Zwecks enthalten, sodass die Nutzenden mit assistiven Technologien sie problemlos verwenden können.

Für die Darstellung der Buttons gibt es eine Kontrastvorgabe über die WCAG 1.4.3, »Kontrast (Minimum)« (Level AA), sowie WCAG 1.4.11, »Nicht-Text-Kontrast« (Level AA). Das Erfolgskriterium 1.4.11 befasst sich mit dem visuellen Kontrast von UI-Elementen (außer Text) und grafischen Objekten. Das Hauptziel besteht darin, einen ausreichenden Kontrast zwischen den visuellen Elementen einer Oberfläche und ihrem Hintergrund zu gewährleisten. Dazu gehören Bedienelemente (z. B. Buttons und Formularfelder), grafische Objekte (z. B. Diagramme und Icons) und Teile von Grafiken, die für das Verständnis des Inhalts oder der Bedienung wesentlich sind. Für grafische Objekte und Benutzeroberflächenelemente gilt ein Mindestkontrastverhältnis von 3 : 1.

Bei der Tastaturnutzung erhält der Button in Abbildung 9.10 den sichtbaren Fokus (WCAG 2.4.7, »Fokus sichtbar«, Level AA), was zur Folge hat, dass auch hier das Kontrastverhältnis von 3 : 1 gegeben sein muss. Abbildung 9.10 zeigt, dass der externe (gelbe) Rahmen des zweiten Buttons nicht kontrastreich zum weißen Hintergrund des (blauen) Buttons ist (Kontrastverhältnis 2 : 1). Dies erfordert eine Anpassung der Farbe, um die Barrierefreiheit zu gewährleisten. Stünde der gelbe Rahmen innerhalb des Buttons mit einem Abstand zum Rand, wäre das Kriterium bestanden. Daher lautet die Empfehlung, zwischen Button und Fokusrahmen eine helle »Lücke« zu lassen, aber darauf zu achten, dass der Kontrast zum Hintergrund ebenfalls ausreichend ist.



Abbildung 9.10 Zu wenig Kontrast bei der Fokussierung des Buttons

Wird die Farbe des Fokus auf Schwarz eingestellt (wie in Abbildung 9.11), ist der Kontrast mit 21 : 1 maximal und damit ausreichend. Für weitere Beispiele siehe <https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html>.



Abbildung 9.11 Kontrastreicher Button bei Fokussierung

Verwenden Sie daher einen Fokusindikator in ausreichender Größe und mit ausreichendem Kontrast. Ein sehr guter Richtwert ist die strenge WCAG 2.4.13, »Erscheinungsbild des Fokus« (Level AAA), die auf das sichtbare Erscheinungsbild des Tastaturfokus abzielt. Ein Bereich des Fokusindikators muss bei aktivem Tastaturfokus ein Kontrastverhältnis von mindestens 3 : 1 aufweisen. Darüber hinaus muss der Fokusindikator eine Umrandung von mindestens 2 CSS-Pixel Dicke um das fokussierte Element aufweisen.

9.10 Fehlerbehandlung in Formularen

Menschen werden Fehler machen. Das lässt sich nicht verhindern. Sie sollten aber erfahren können, was sie im Falle eines Fehlers tun können oder, besser noch, was sie tun können, um Fehler zu vermeiden. Abbildung 9.12 zeigt eine Fehlermeldung,

sofern der gewählte Nutzername bereits vorhanden ist, sowie eine Fehlermeldung bei einem Passwort ohne Zahl, bei dem laut Passwortregel eine enthalten sein muss.

Das Passwort sollte mindestens sechs Zeichen und mindestens eine Zahl (0-9) enthalten

Passwort

Fehlermeldungen zeigen an, was das Problem ist und wie man es beheben kann

Nutzername Passwort

Der Nutzername Lilly1234 ist bereits vergeben. Das Passwort muss mindestens eine Zahl enthalten

Abbildung 9.12 Fehlermitteilung nebst Behebung

Die 10 Heuristiken von Nielsen und Molich (<https://www.nngroup.com/articles/ten-usability-heuristics/>) befassen sich bereits seit 1994 aus Usability-Sicht mit der Bedeutung von Fehlern bei der Nutzung von Websites und dem Umgang damit. In Usability-Tests wird immer wieder festgestellt, dass z. B. Kreditkartennummern in Viererblöcken (wie auf der Kreditkarte eingeprägt) eingegeben werden. Ebenso werden bei Telefonnummern Vorwahl und Rufnummer durch einen Schrägstrich getrennt oder mit vorangestelltem +49 eingegeben (also Zeichen statt ausschließlich Ziffern). Im Idealfall »verzeiht« das Formular diese Eingaben und verarbeitet die Daten entsprechend.

Kontextuelle Hilfe in Formularen bietet Nutzenden proaktiv zusätzliche Informationen oder Anleitungen zum Ausfüllen eines Formularfeldes. Diese Hilfe kann in verschiedenen Formen angeboten werden, z. B. als Tooltips, als beschreibender Text direkt unter den Eingabefeldern oder als Hilfesymbol, das weitere Informationen anzeigt, wenn die Nutzenden mit der Maus darüberfahren oder daraufklicken. Die kontextuelle Hilfe ist für die Barrierefreiheit und Benutzerfreundlichkeit besonders wichtig, da sie hilft, Eingabefehler zu vermeiden und den Nutzenden klarzumachen, welche Art von Information erwartet wird.

WCAG 3.3.5, »Hilfe« (Level AAA), verlangt, dass den Nutzenden bei Bedarf Hilfe zur Verfügung gestellt wird. Das kann durch verschiedene Mechanismen erfolgen, einschließlich kontextueller Hilfe. Das Ziel besteht darin, sicherzustellen, dass alle Nutzenden Formulare erfolgreich ausfüllen können, indem sie Zugang zu notwendigen Informationen erhalten, wann immer es nötig ist.

Listing 9.8 zeigt die Anforderungen an ein Passwort in einem Registrierungsformular. Kontextuelle Hilfe könnte verwendet werden, um die spezifischen Anforderungen an das Passwort zu erläutern (mindestens 8 Zeichen etc.).

```
<label for="password">Passwort</label>
<input type="password" id="password" name="password" aria-describedby=
"passwordHelp">
<div id="passwordHelp" class="help-text">
    Ihr Passwort muss mindestens 8 Zeichen lang sein, einen Großbuchstaben,
    eine Zahl und ein Sonderzeichen enthalten.
</div>
```

Listing 9.8 Hinweis zu den Anforderungen des Passwortes

In Listing 9.8 wird das `aria-describedby`-Attribut verwendet, um das Passwordeingabefeld mit einem Beschreibungstext (`<div id="passwordHelp" class="help-text">`) zu verknüpfen. Dieser Text erläutert die Anforderungen an die Passwordeingabe. Für die Nutzenden von Screenreadern wird dieser Beschreibungstext nach dem Fokussieren auf das Passwortfeld ausgegeben, was ihnen sofortigen Zugriff auf die Anforderungen bietet. Zusätzlich können Links auf Hilfeseiten angeboten werden, die gegebenenfalls sehr ausführliche Anleitungen bieten (z. B. zum Ausfüllen des Formulars zur neuen Bemessungsgrundlage der Grundsteuer). Natürlich muss dieser Hilfetext mittels CSS an der richtigen Stelle im Layout platziert und richtig formatiert werden.

WCAG 3.3.3, »Fehlerempfehlung« (Level AA), fordert das Angebot eines Vorschlags zur Korrektur des Fehlers, wenn ein automatischer Fehler bei der Eingabe erkannt wird und die Korrektur des Fehlers bekannt ist – es sei denn, dies würde die Sicherheit oder den Zweck des Inhalts gefährden. Google zeigt visuell passende Vorschläge (früher »Meinten Sie ...«) an, wenn die Nutzenden das Wort falsch geschrieben haben und bietet zusätzlich die Möglichkeit, nach der vermeintlich falschen Schreibweise zu suchen. Ziel ist es, den Nutzenden zu helfen, Fehler in ihren Eingaben zu korrigieren, indem konkrete Vorschläge zur Behebung des Problems angeboten werden. Dies trägt dazu bei, Frustrationen zu vermeiden, die Benutzerfreundlichkeit zu verbessern und sicherzustellen, dass alle Nutzenden erfolgreich mit Webinhalten interagieren können. Die Praxis der Bereitstellung von Fehlerhinweisen und Korrekturvorschlägen ist für die Barrierefreiheit besonders wichtig, weil es den Nutzenden ermöglicht, Fehler zu korrigieren, ohne raten zu müssen, was falsch ist. Geben Sie Rückmeldungen zu Interaktionen, indem Sie beispielsweise die erfolgreiche Formularübermittlung bestätigen, die Nutzenden auf Fehler hinweisen oder über Änderungen auf der Seite informieren. Anweisungen sollten leicht erkennbar sein. Wichtige Rückmeldungen, die eine Aktion der Nutzenden erfordern, sollten auffällig gestaltet werden (siehe Abbildung 9.12).

Bitte korrigiere die folgenden Fehler

1. Die E-Mail-Adresse ist ungültig
2. Eine Telefonnummer ist erforderlich

Bitte ergänze deine Daten

Mit * markierte Felder sind Pflichtfelder.

Name

Email*

Die E-Mail-Adresse ist ungültig

Geburtsdatum

Telefonnummer*

Eine Telefonnummer ist erforderlich

Abbildung 9.13 Kommunikation über und Darstellung eines Eingabefelders
(Quelle/weitere Tipps: <https://www.w3.org/WAI/tips/designing>)

Es ist unvermeidbar, dass die Nutzenden bei der Eingabe Fehler machen (Murphys Gesetz!) und z. B. Pflichtfelder übersehen oder Daten im »falschen« Format eingeben (Geburtsdatum mit vierstelliger Jahreszahl). Daher ist sowohl aus Usability- als auch aus Sicht der Barrierefreiheit wichtig, dass die Nutzenden auf die Fehler aufmerksam gemacht werden und die Fehlermeldung in den aktiven Fokus gerückt wird. Die Fehlermeldung sollte eine konkrete Abhilfe beinhalten (»Eine Telefonnummer ist erforderlich«) und nicht nur einen allgemeinen Hinweis (»Das ist ein Pflichtfeld«). Hilfreich sind sogenannte Inline-Validierungen, d. h., jedes Feld wird unmittelbar nach der Eingabe überprüft und gegebenenfalls visuell mit einem grünen Häkchen bestätigt. So wird zum einen die Eingabe positiv unterstützt (virtuelles »Schulterklopfen«) und zum anderen findet keine demotivierende Fehleraggregation nach Absenden des Formulars statt. WCAG 3.3.1, »Fehlererkennung« (Level A), fordert, dass bei der automatischen Erkennung eines Eingabefelders den Nutzenden der Fehler deutlich gemacht und die Fehlerstelle (»Anrede fehlt«) mitgeteilt wird. Darüber hinaus soll Nutzenden nach Möglichkeit ein Vorschlag zur Behebung des Fehlers gemacht werden (WCAG 3.3.3, »Fehlerempfehlung«, Level AA). Damit sollen die Nutzenden sofort verstehen, wenn ein Fehler in den von ihnen eingegebenen Daten vorliegt, worin genau dieser Fehler besteht und wie sie ihn korrigieren können.

Abbildung 9.14 zeigt die Umsetzung der WCAG 3.3.1 in einem Anmeldeformular, in das die Nutzenden ihre E-Mail-Adresse und ihr Passwort eingeben müssen. Wenn sie eine ungültige E-Mail-Adresse eingeben und versuchen, das Formular abzusenden, wird eine Fehlermeldung angezeigt, die das Problem klar identifiziert und einen Vorschlag zur Korrektur anbietet.

The screenshot shows a login form with two input fields: "E-Mail:" containing the text "test" and "Passwort:". To the right of the password field is a button labeled "Anmelden". Below the form, a red error message box is displayed with a white exclamation mark icon. The message reads: "Die E-Mail-Adresse muss ein @-Zeichen enthalten. In der Angabe 'test' fehlt ein @-Zeichen."

Abbildung 9.14 Fehlermeldung bei Prüfung auf fehlendes @-Zeichen

Listing 9.9 zeigt den Code für das Formular einschließlich der Gültigkeitsprüfung.

```
<form id="loginForm">
  <label for="email">E-Mail:</label>
  <input type="email" id="email" name="email" aria-describedby="emailError"
required>
  <span id="emailError" class="error" style="display: none;" role=
"alert">Bitte geben Sie eine gültige E-Mail-Adresse ein.</span>

  <label for="password">Passwort:</label>
  <input type="password" id="password" name="password" required>

  <button type="submit">Anmelden</button>
</form>

<script>
document.getElementById('loginForm').addEventListener('submit',
function(event) {
  var email = document.getElementById('email');
  var error = document.getElementById('emailError');

  // Überprüfung auf E-Mail-Gültigkeit
  var emailPattern = /^[^s@]+@[^s@]+\.[^s@]+$/; // Regulärer Ausdruck für
einfache E-Mail-Validierung
  if (!emailPattern.test(email.value)) {
    error.style.display = 'block'; // Fehlermeldung anzeigen
    email.setAttribute('aria-invalid', 'true'); // Markiere das Feld als
ungültig
    email.focus(); // Fokus auf das fehlerhafte Feld setzen
    event.preventDefault(); // Verhindert das Absenden des Formulars
  } else {
```

```

        error.style.display = 'none'; // Fehlermeldung ausblenden
        email.removeAttribute('aria-invalid'); // Entferne die
Ungültigkeitsmarkierung
    }
});
</script>
<style>
/* Verbesserte Darstellung von Fehlermeldungen */
.error {
    color: red;
    font-size: 0.9em;
}
</style>

```

Listing 9.9 Fehlermeldung bei Prüfung auf fehlendes @-Zeichen

Leider validieren noch nicht alle Browser (Safari) automatisch bei `input type="email"`. In Listing 9.9 wird deshalb JavaScript verwendet, um unter anderem zu prüfen, ob die E-Mail-Adresse ein @-Zeichen enthält. Wenn die E-Mail-Adresse ungültig ist, wird das Absenden des Formulars verhindert, ein Fehlerhinweis erscheint und die Nutzenden erhalten eine klare Anweisung, was korrigiert werden muss. Die Fehlermeldung ist dabei direkt mit dem Eingabefeld verknüpft, sodass das Problem leicht lokalisiert werden kann.

Besonders wichtig ist die Reduzierung von Fehlern, die erhebliche negative Auswirkungen auf die Nutzenden haben. »Aus Versehen« ein langfristiges Abonnement abgeschlossen zu haben, ist nicht nur ärgerlich, sondern kann auch schwerwiegende finanzielle Folgen haben. Daher zielt die WCAG 3.3.4, »Fehlervermeidung (Rechtlich, Finanziell, Daten)« (Level AA), darauf ab, zusätzliche Unterstützung zu geben und die Möglichkeit zu bieten, Eingaben in Formularen zu überprüfen, zu korrigieren und zu bestätigen, bevor endgültige Aktionen durchgeführt werden, die rechtliche oder finanzielle Konsequenzen haben können oder die persönlichen Daten betreffen. Damit sollen Fehler minimiert und den Nutzenden mehr Kontrolle über ihre Aktionen gegeben werden. Daher gilt mindestens eine der folgenden Bestimmungen:

1. Reversibel: Versendete Daten sind umkehrbar.
2. Geprüft: Von den Nutzenden eingegebene Daten werden auf Eingabefehler hin überprüft und sie erhalten die Möglichkeit, diese zu korrigieren.
3. Bestätigt: Es gibt einen Mechanismus, Formulardaten zu überprüfen, zu bestätigen und zu korrigieren, bevor sie endgültig abgesendet werden.

Stellen Sie sich ein Onlinebanking-Formular vor, mit dem die Nutzenden Geldbeträge überweisen können. In diesem Prozess wird man nach dem Absenden des Formulars

auf eine Seite geleitet, auf der alle Eingaben aufgeführt und zur Überprüfung angezeigt werden, bevor die Nutzenden die Möglichkeit haben, die Transaktion endgültig zu bestätigen oder zurückzugehen und Fehler zu korrigieren. Diese Vorgehensweise ist besonders wichtig für diejenigen, bei denen die Gefahr besteht, dass sie Fehler beim Ausfüllen von Formularen übersehen, z. B. ältere Menschen, Menschen mit kognitiven Einschränkungen oder die Nutzenden von assistiven Technologien.

Die strengere WCAG 3.3.6, »Fehlervermeidung (Alle)« (Level AAA), betont die Bedeutung der Fehlervermeidung für alle Arten von Benutzereingaben in Formularen oder bei der Übermittlung von Daten im Allgemeinen. Ziel ist es, sicherzustellen, dass Menschen vor potenziellen Fehlern geschützt werden, indem der Prozess redundante Überprüfungen anbietet und allen die Möglichkeit gibt, Eingaben zu widerrufen und zu korrigieren, bevor sie endgültig abgeschickt werden. WCAG 3.3.6 geht über die Anforderungen von WCAG 3.3.4 hinaus, da sie sich auf alle Benutzereingaben und -interaktionen bezieht und nicht nur auf die rechtlich und finanziell relevanten.

Die Reduzierung von Fehleingaben ist für eine effiziente und benutzerfreundliche Erfahrung entscheidend. Fehleingaben können Frustration verursachen, zu Abbrüchen führen und die Datenqualität beeinträchtigen. Die folgenden Best Practices sollen dazu beitragen, Fehleingaben zu minimieren und die Benutzerinteraktion mit Formularen zu verbessern:

- ▶ Klare und eindeutige Beschriftungen: Verwenden Sie eindeutige, präzise Beschriftungen für jedes Eingabefeld. Stellen Sie sicher, dass die Nutzenden verstehen, welche Informationen angefordert werden. Verwenden Sie sprachlich einfache Anweisungen.
- ▶ Sinnvoller Einsatz von Platzhaltern: Verwenden Sie Platzhaltertexte, um Beispiele oder das erwartete Format (z. B. für Datumsangaben) zu illustrieren. Beachten Sie jedoch, dass Platzhalter nicht als Ersatz für Beschriftungen (Labels) dienen sollten. Außerdem ist auf die Mindestkontraste zu achten.
- ▶ Fehlermeldungen und Validierung: Implementieren Sie eine Echtzeitvalidierung mit sofortigem Feedback. Klare und verständliche Fehlermeldungen helfen den Nutzenden, Eingabefehler schnell zu identifizieren und an der richtigen Stelle zu korrigieren. Stellen Sie sicher, dass Fehlermeldungen spezifisch sind und Vorschläge zur Fehlerbehebung enthalten.
- ▶ Hilfetexte und Tipps: Bieten Sie Hilfetexte oder Tipps für komplexere Eingabefelder an. Diese können zusätzliche Informationen oder Anleitungen zum Ausfüllen des Formulars enthalten.
- ▶ Autovervollständigung: Nutzen Sie standardkonforme Möglichkeiten der Autovervollständigung (z. B. `autocomplete`), um die Eingabe häufig verwendeter, personenbezogener Daten wie Namen, Adressen oder E-Mail-Adressen zu beschleunigen und Fehler zu reduzieren. Achten Sie darauf, dass das Label sichtbar ist. Bei

automatischem Browser-Fill-In von Informationen muss dem Nutzenden klar sein, welche Informationen in dem Feld erwartet werden.

- ▶ **Bestätigung der Eingaben:** Bei kritischen oder ungewöhnlichen Datenabfragen bitten Sie die Nutzenden, ihre Eingaben zu bestätigen (z. B. durch erneute Eingabe von E-Mail-Adressen oder Passwörtern). Eine abschließende Übersicht der Daten (z. B. in Form einer Bestellübersicht) verringert Fehleingaben.
- ▶ **Mobilfreundliche Formulare:** Stellen Sie sicher, dass Formulare auf mobilen Endgeräten gut nutzbar sind, indem Sie responsive Designs verwenden. Größere Formularelemente und ausreichende Abstände zwischen den Feldern können die Eingabe auf Touchscreens erleichtern.
- ▶ **Begrenzte Eingabeoptionen:** Vermeiden Sie zu komplexe Auswahlmöglichkeiten. Verwenden Sie Dropdown-Menüs, Radiobuttons oder Checkboxes, um die Anzahl der möglichen Eingaben zu begrenzen und Fehler zu minimieren.
- ▶ **Benutzerführung und Fortschrittsanzeige:** Verwenden Sie bei längeren Formularen eine Fortschrittsanzeige und gliedern Sie das Formular in logische Abschnitte. Dies hilft den Nutzenden, den Überblick zu behalten und das Formular Schritt für Schritt auszufüllen. Vor dem Löschen oder Zurücksetzen der eingegebenen Informationen sollte erneut die Zustimmung eingeholt werden.

9.11 Spezifische Eigenschaften der Eingabefelder

Neben der Grundstruktur sind bei einzelnen Eingabefeldern spezifische Aspekte der Barrierefreiheit und Usability zu berücksichtigen.

9.11.1 Dropdown-Felder und Eingabe von Mehrfachauswahlen

Eine Besonderheit in Formularen sind Dropdown-Felder. Sie ermöglichen es Nutzenden, eine Auswahl aus einer vordefinierten Menge zu treffen. Ein `<select>`-Element in HTML-Formularen ermöglicht die Auswahl aus mehreren vordefinierten Optionen und enthält mehrere `<option>`-Elemente, die die verfügbaren Optionen repräsentieren, aus denen die Nutzenden auswählen können. Optional kann ein `<optgroup>`-Element verwendet werden, um verwandte Optionen in Gruppen zu unterteilen. Abbildung 9.15 zeigt ein Formular eines Lieferservice, in dem die Nutzenden ihr Lieblingsobst auswählen sollen, wobei sie genau ein Obst auswählen können (Code siehe Listing 9.10).

```
<label for="frucht">Wählen Sie Ihr Lieblingsobst:</label>
<select id="frucht" name="frucht" required>
  <option value="">--Bitte wählen--</option>
  <optgroup label="Kernobst">
```

```

    <option value="apple">Apfel</option>
    <option value="pear">Birne</option>
</optgroup>
<optgroup label="Steinobst">
    <option value="peach">Pfirsich</option>
    <option value="plum">Pflaume</option>
</optgroup>
</select>

```

Listing 9.10 Definition eines Dropdown-Feldes mit einer Auswahl

Die erste `<option>` im `<select>`-Menü sollte eine Handlungsaufforderung oder eine Anleitung sein, wie z. B. »Bitte wählen«. Diese Option kann leer sein (`value=""`) und sollte in der Regel so eingestellt sein, dass sie nicht auswählbar ist, wenn bereits andere Optionen ausgewählt wurden. Wenn das Auswahlmenü viele Optionen enthält, kann das `<optgroup>`-Element verwendet werden, um diese in logische Gruppen zu unterteilen. Jede `<optgroup>` hat ein `label`-Attribut, das die Gruppe beschreibt. Dies kann die Navigation innerhalb der Liste für die Nutzenden mit Screenreadern erleichtern, da die Struktur des Menüs klarer wird.

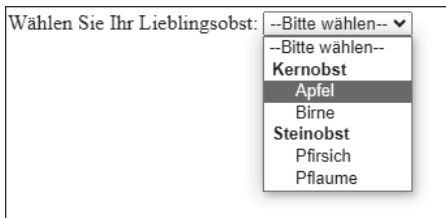


Abbildung 9.15 Auswahl von Alternativen

Eine Mehrfachauswahlliste (oder Multiselect-Liste) in einem HTML-Formular erweitert die Funktionalität des klassischen `<select>`-Elements, indem sie es den Nutzenden ermöglicht, mehr als eine Option aus der Liste auszuwählen. Diese Art von Auswahl ist in Situationen nützlich, in denen eine einfache Auswahl (Single Select) nicht ausreicht, weil die Nutzenden mehrere Elemente aus einer Reihe von Optionen wählen müssen oder dürfen. Die Implementierung und die Aspekte der Barrierefreiheit unterscheiden sich etwas von denen eines Single-Select-`<select>`-Elements. Abbildung 9.16 zeigt die Möglichkeit der Mehrfachauswahl von Obstsorten (Code siehe Listing 9.11).

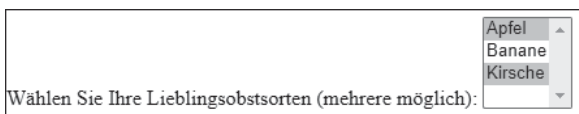


Abbildung 9.16 Mehrfachauswahl im Feld

```

<label for="fruits">Wählen Sie Ihre Lieblingsobstsorten (mehrere möglich):
</label>
<select id="fruits" name="fruits" multiple>
  <option value="apple">Apfel</option>
  <option value="banana">Banane</option>
  <option value="cherry">Kirsche</option>
  <!-- Weitere Optionen -->
</select>

```

Listing 9.11 Mehrfachauswahl in einer Dropdown-Liste

Die Nutzenden können mehrere Optionen gleichzeitig auswählen, indem sie die **Ctrl**/**Strg**- (Windows/Linux) oder **Cmd**-Taste (Mac) gedrückt halten, während sie auf die Optionen klicken. Bei der reinen Tastaturbedienung muss man dafür die **Ctrl**/**Strg**-Taste (oder **Cmd**) gedrückt halten, während man mit den Pfeiltasten zu den Elementen geht, die dann mit der Leertaste ausgewählt werden können. Durch das Hinzufügen des Attributs `multiple` zum `<select>`-Element wird dem Browser (und den Nutzenden) signalisiert, dass Mehrfachauswahlen möglich sind. Durch die zusätzliche Tastenkombination ist dies aus Sicht der Barrierefreiheit nicht optimal, da neben der Auswahl gleichzeitig die **Strg**/**Cmd**-Taste gehalten werden und gegebenenfalls der Modus im Screenreader angepasst werden muss.

9.11.2 Drag-and-Drop-Aktionen

WCAG 2.5.7, »Ziehbewegungen« (Level AA), befasst sich mit alternativen Methoden für Drag-and-Drop-Aktionen, die nicht auf komplexen Bewegungen oder Gesten basieren. Webinhalte sollten so gestaltet sein, dass Funktionen, die normalerweise durch Ziehen erreicht werden, auch durch einfaches Klicken oder Tastatureingaben genutzt werden können. Dies wird z. B. in WordPress durch den zusätzlichen Button in Abbildung 9.17 umgesetzt.

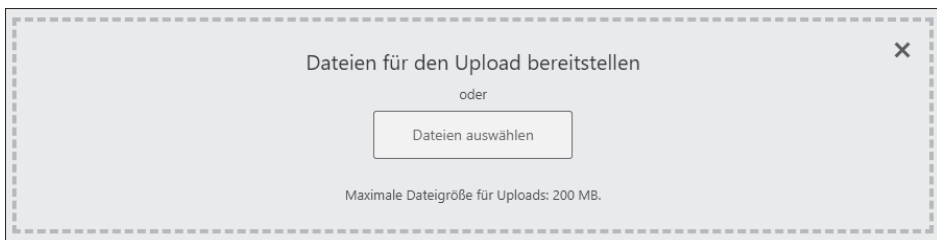


Abbildung 9.17 Upload von Dateien in WordPress per Ziehbewegung und Button

Die Usability des Datei-Uploads in WordPress ist auf den ersten Blick in Abbildung 9.17 durch eine einfache Ziehbewegung (Drag and Drop) ausreichend sichergestellt.

Das heißt, Sie ziehen »einfach« die Datei in den Upload-Bereich, und schon wird die Datei hochgeladen. Dasselbe gilt z. B. für intuitive Gestaltungslösungen wie Schiebeschalter (Ein/Aus) bzw. Zeigerelemente wie eine Preisspanne von – bis auf Hotelbuchungsseiten. Unter dem Gesichtspunkt der Barrierefreiheit muss der Upload auch für Menschen mit motorischen Einschränkungen möglich sein, die Schwierigkeiten haben, eine Maus zu steuern oder eine Berührungsgeste auf einem Touchscreen auszuführen. Gegebenenfalls wird die Webseite über Sprachbefehle gesteuert.

9.11.3 Redundante Eingaben

WCAG 3.3.7, »Redundante Eingabe« (Level AA), zielt darauf ab, die Nutzenden davor zu bewahren, Informationen innerhalb eines Vorgangs erneut eingeben zu müssen, wenn sie diese Informationen bereits einmal eingegeben haben. Das ist besonders wichtig für die Nutzenden mit kognitiven, sprachlichen und/oder lernbezogenen Behinderungen, die möglicherweise Schwierigkeiten haben, sich an Informationen zu erinnern (insbesondere Kurzzeitgedächtnis) oder diese bei jeder Nutzung erneut einzugeben. Der Ausfüllprozess soll dadurch vereinfacht werden, es sei denn, dass die mehrfache Eingabe aus Sicherheitsgründen notwendig ist (z. B. bei der Passworteingabe).

Ein häufiges Beispiel für die Anwendung dieser Richtlinie ist ein Onlinebestellformular. Stellen Sie sich vor, die Nutzenden müssen während eines Bestellvorgangs zuerst die Lieferadresse und dann die Rechnungsadresse eingeben. Um die Anforderungen von WCAG 3.3.7 zu erfüllen, könnte das Formular die Option enthalten, die Lieferadresse in die Rechnungsadresse zu übernehmen, wie in Abbildung 9.18 dargestellt.

The image shows a form with two sections: 'Lieferadresse' and 'Rechnungsadresse'. In the 'Lieferadresse' section, there are input fields for 'Name' (containing 'Tobias') and 'Adresse' (containing 'Würzburg'). In the 'Rechnungsadresse' section, there is a checked checkbox labeled 'Lieferadresse als Rechnungsadresse verwenden', followed by input fields for 'Name' (containing 'Tobias') and 'Adresse' (containing 'Würzburg'). At the bottom of the form is a 'Bestellen' button.

Abbildung 9.18 Optionale Übernahme der Lieferadresse

Die Nutzenden könnten das Kontrollkästchen aktivieren, um die bereits eingegebene Lieferadresse automatisch in das Feld Rechnungsadresse zu kopieren. Listing 9.12 zeigt den entsprechenden Code mit dem notwendigen onclick-Ereignis.

```
<!DOCTYPE html>
<html lang="de">
<head>
```

```
<meta charset="UTF-8">
<title>Bestellformular</title>
<script>
  function copyAddress() {
    if (document.getElementById('sameAddress').checked) {
      document.getElementById('billingName').value =
document.getElementById('shippingName').value;
      document.getElementById('billingAddress').value =
document.getElementById('shippingAddress').value;
      // Weitere Felder kopieren...
    }
  }
</script>
</head>
<body>
  <form>
    <fieldset>
      <legend>Lieferadresse</legend>
      <label for="shippingName">Name:</label>
      <input type="text" id="shippingName" name="shippingName"
required><br>
      <label for="shippingAddress">Adresse:</label>
      <input type="text" id="shippingAddress" name="shippingAddress"
required><br>
    </fieldset>

    <fieldset>
      <legend>Rechnungsadresse</legend>
      <input type="checkbox" id="sameAddress" name="sameAddress"
onclick="copyAddress()">
      <label for="sameAddress">Lieferadresse als Rechnungsadresse
verwenden</label><br>
      <label for="billingName">Name:</label>
      <input type="text" id="billingName" name="billingName"
required><br>
      <label for="billingAddress">Adresse:</label>
      <input type="text" id="billingAddress" name="billingAddress"
required><br>
    </fieldset>

    <input type="submit" value="Bestellen">
  </form>
</body>
</html>
```

```

    </form>
</body>
</html>

```

Listing 9.12 Übernahme der Adressen – einmaliges Eingeben

Die Richtlinie WCAG 3.3.7 ermöglicht einen effizienten und flüssigen Prozess, der es den Nutzenden erlaubt, Aufgaben schneller und mit weniger Aufwand zu erledigen. Darüber hinaus reduziert sie die kognitive Last und potenzielle Frustration für alle Nutzenden, insbesondere für diejenigen mit Gedächtnis- oder Konzentrationschwierigkeiten.

9.11.4 Authentifizierung leicht gemacht

WCAG 3.3.8, »Zugängliche Authentifizierung (Minimum)« (Level AA), bezieht sich auf die Möglichkeit für die Nutzenden mit unterschiedlichen Fähigkeiten, sich sicher und effizient anzumelden oder zu authentifizieren. Authentifizierungsverfahren, wie z. B. die Anmeldung mit Benutzername und Passwort, sollten so gestaltet sein, dass sie für Menschen mit Behinderungen zugänglich sind, ohne dass sie kognitive Funktionen oder physische Fähigkeiten benötigen, die über das übliche Maß hinausgehen. Es sollten alternative Authentifizierungsmethoden angeboten werden, die keine komplexe Gedächtnisleistung oder komplexe Berechnungen erfordern.

Ein gutes Beispiel, das WCAG 3.3.8 entspricht, ist ein zweistufiges Authentifizierungsverfahren, das neben einem herkömmlichen Passwort auch die Möglichkeit bietet, sich über einen (Magic-)Link in einer E-Mail oder per SMS zu authentifizieren. Dadurch wird es Menschen, die Schwierigkeiten mit visuellen Captchas oder komplexen Passwörtern haben, möglich, sich dennoch relativ sicher zu authentifizieren.

WCAG 3.3.9, »Zugängliche Authentifizierung (erweitert)« (Level AAA), geht restriktiver vor. Dieses Kriterium stellt sicher, dass kein Schritt im Authentifizierungsprozess einen kognitiven Funktionstest erfordert, es sei denn, es wird eine alternative Methode oder Unterstützung angeboten. Dieses Erfolgskriterium soll sicherstellen, dass es eine zugängliche, einfach zu bedienende und sichere Methode gibt, sich anzumelden, auf Inhalte zuzugreifen und Aufgaben auszuführen. Dieses Kriterium entspricht WCAG 3.3.8, jedoch ohne die Ausnahmen für Objekte und Inhalte, die vom Nutzenden bereitgestellt werden.

Jeder erforderliche Schritt des Authentifizierungsprozesses darf demnach

- ▶ weder eine Auswahl von Bildern, Videos oder Audioclips anzeigen, bei denen die Nutzenden das von ihnen bereitgestellte Bild auswählen müssen,
- ▶ noch eine Auswahl von Bildern anzeigen, bei denen die Nutzenden die Bilder auswählen müssen, die ein bestimmtes Objekt enthalten, z. B. ein Auto.

9.12 Hilfestellungen

Typischerweise benutzen wir Menschen eine Website oder eine Anwendung allein, niemand schaut zu und hilft bei Problemen. Bei Usability-Tests fällt manchmal auf, dass bestimmte Aspekte nicht verstanden werden (Fachbegriffe, mehrdeutige Interpretationsmöglichkeiten etc.) oder dass Abläufe unklar sind. Aus Usability-Sicht ist es am besten, wenn das System keine zusätzlichen Erklärungen benötigt. Es kann jedoch notwendig sein, eine Dokumentation bereitzustellen, damit die Nutzenden verstehen, wie sie ihre Aufgaben erledigen können (10. Heuristik »Hilfe und Dokumentation« von Niensens Usability-Heuristiken für die Gestaltung von Benutzeroberflächen)¹. Also nennen Sie nie allein die Fehlermeldung, sondern geben Sie immer auch Korrekturhilfen.

WCAG 3.3.5, »Hilfe« (Level AAA), greift das Thema Hilfe ebenfalls auf und fordert kontextsensitive Hilfe, damit die Nutzenden Fehler vermeiden können. Mit kontextsensitiver Hilfe kann herausgefunden werden, wie eine Operation durchgeführt werden kann, ohne den Überblick zu verlieren. Kontextsensitive Hilfe sollte nur dann angeboten werden, wenn die Beschriftung nicht ausreicht, um alle Funktionen verständlich zu beschreiben. Das Vorhandensein von kontextsensitiver Hilfe sollte für alle Nutzenden offensichtlich sein; sie sollte verfügbar sein, wann immer sie benötigt wird.

Insbesondere für Formulare sollte Folgendes gegeben sein:

- ▶ Unterstützung durch Assistenten wie Chatbots (Diese müssen selbst barrierefrei sein.)
- ▶ eine Rechtschreibprüfung und Vorschläge für Texteingaben
- ▶ Textanweisungen am Anfang eines Formulars oder eine Reihe von Feldern, die die erforderlichen Eingaben beschreiben
- ▶ das erwartete Datenformat im Fall von File-Uploads und entsprechende Beispiele

9.13 Zusammenfassung des Kapitels

Hier sind noch einmal die wichtigsten Punkte dieses Kapitels im Überblick:

- ▶ Verwenden Sie semantisch korrekte HTML-Strukturen wie `<form>`, `<input>`, `<label>`, `<fieldset>` und `<legend>`.
- ▶ Wählen Sie passende `<input>`-Typen (z. B. `type="email"`) für die erwarteten Daten, um mobile Tastaturen optimal anzupassen.

¹ <https://www.nngroup.com/articles/ten-usability-heuristics/>

- ▶ Verwenden Sie für jedes Eingabefeld ein sichtbares `<label>`, das mit dem `for`-Attribut an das entsprechende Feld gebunden ist. Platzieren Sie Labels in unmittelbarer Nähe zu ihren Eingabefeldern für eine bessere Verständlichkeit.
- ▶ Nutzen Sie ARIA-Attribute wie `aria-label` oder `aria-labelledby` für Felder ohne sichtbare Labels, um die Zugänglichkeit sicherzustellen. Kombinieren Sie unsichtbare Beschriftungen mit CSS, um visuelle Designs zu erhalten, ohne die Barrierefreiheit zu beeinträchtigen.
- ▶ Ordnen Sie Formularfelder in einer logischen und intuitiven Reihenfolge an, die dem natürlichen Nutzungsfluss entspricht. Achten Sie auf eine konsistente Tab-Reihenfolge zur Unterstützung der Tastaturnavigation.
- ▶ Nutzen Sie das `required`-Attribut für Pflichtfelder, um eine native Browservalidierung und Fehlermeldungen zu aktivieren.
- ▶ Deaktivieren Sie die Autokorrektur bei Eingabefeldern wie E-Mail-Adressen. Vermeiden Sie dadurch unerwünschte automatische Anpassungen, die zu Eingabefehlern führen können.
- ▶ Implementieren Sie das `autocomplete`-Attribut mit spezifischen Werten (z. B. `autocomplete="given-name"`), um die Autovervollständigung bei personenbezogenen Angaben zu unterstützen.
- ▶ Stellen Sie einen ausreichenden Kontrast (mindestens 3 : 1) für Buttons und Fokusindikatoren sicher. Sorgen Sie für sichtbare Fokuszustände bei Tastaturnavigation und für klare Beschriftungen der Buttons.
- ▶ Bieten Sie eine Echtzeitvalidierung und sofortiges Feedback bei Eingabefehlern. Geben Sie klare, spezifische Fehlermeldungen mit Korrekturvorschlägen.
- ▶ Ermöglichen Sie Nutzenden, Eingaben zu überprüfen und zu korrigieren, besonders bei kritischen Aktionen (WCAG 3.3.4 und 3.3.6).
- ▶ Bieten Sie alternative Bedienmöglichkeiten zu Drag and Drop, z. B. durch zusätzliche Buttons. Stellen Sie sicher, dass alle Funktionen auch ohne komplexe Gesten oder Bewegungen nutzbar sind (WCAG 2.5.7).
- ▶ Vermeiden Sie doppelte Dateneingaben, indem Sie Optionen zur Datenübernahme anbieten (z. B. Checkbox für gleiche Rechnungsadresse).
- ▶ Implementieren Sie zugängliche Authentifizierungsmethoden ohne komplexe kognitive Anforderungen. Bieten Sie Alternativen zu Passwörtern und Captchas, wie z. B. Magic Links oder biometrische Verfahren (WCAG 3.3.8 und 3.3.9).
- ▶ Stellen Sie kontextsensitive Hilfe bereit, z. B. durch Hilfetexte, Tooltips oder Chatbots.
- ▶ Bieten Sie klare Anweisungen und Beispiele für komplexe Eingaben oder Formate. Unterstützen Sie Nutzende durch Rechtschreibprüfungen und Eingabevorschläge, um Fehler zu vermeiden.