

Quick-Check:

Ist unsere Website barrierefrei?

Gültig ab 2025 – besonders relevant für Tourismusbetriebe

Nach den Prinzipien: **Wahrnehmbar, Bedienbar, Robust** lt. WCAG

1. Texte verständlich?

- ☐ Sind Inhalte klar strukturiert (Überschriften, Absätze)?
 - ✂ Was zusammengehört, muss auch im Code zusammengehören. Strukturelle und inhaltliche Beziehungen müssen programmatisch erkennbar sein – also auch für Screenreader und andere Hilfsmittel verständlich.

Erfolgskriterium: **Textabstand**

 Typografie ist nur barrierefrei, wenn sie sich entfalten darf. Benutzer müssen Zeilenhöhe, Absatzabstand, Zeichenabstand und Wortabstand anpassen können, ohne dass Inhalte unlesbar werden.

- ☐ Wird auf komplizierte Fachbegriffe verzichtet?
- ☐ Gibt es Alternativen zu rein visuellen Infos (siehe „Erfolgskriterium Nicht-Text-Inhalt“)?

Und:  Wenn du's lesen kannst, mach's als Schrift – nicht als Bild.

Vermeide Text in Bildern, wenn der gleiche visuelle Effekt auch mit echtem Text und CSS erreicht werden kann, weil Bilder von Text nicht skaliert, vorgelesen oder angepasst werden können – echter Text schon.

Erfolgskriterium **Nicht-Text-Inhalt**

 Alles, was kein Text ist, braucht Text – oder wird bewusst ignoriert.

Textalternativen sind also erforderlich für:

- Bilder, Icons, Symbole → kurze Beschreibung (ca. 100 Zeichen)
- Komplexe Grafiken (z. B. Diagramme, Infografiken)
 - ausführliche Beschreibung (bis zu 200 Wörter), z. B. als Link oder direkt darunter
- Steuerelemente & Formulare (z. B. Buttons, Icons)
 - klare textuelle Funktion (z. B. aria-label="Löschen")
- Zeitbasierte Medien (z. B. Videos, Livestreams, Audios)
 - kurze Beschreibung mit Zweck und Inhalt
- CAPTCHAs → erklärender Text + barrierefreie Alternative (z. B. Audio-CAPTCHA)

Keine Textalternative (aber korrekt gekennzeichnet) ist notwendig für:

- Dekorative Inhalte → mit alt="" oder nur via CSS einbinden
- HTML/ARIA/Sonstiges-Beispiel – Kurz & Einfach:


```

```

2. Bilder mit Alternativtext?

- ☐ Haben alle Bilder einen sogenannten Alt-Text, der beschreibt, was zu sehen ist (siehe oben)?
- ☐ Gibt es bei wichtigen Grafiken (z. B. Lageplan) eine textliche Beschreibung?

Erfolgskriterium: **Nicht-Text-Kontrast**

☞ Auch was kein Text ist, muss sichtbar sein.

Bedienelemente und grafische Objekte haben ein Kontrastverhältnis von mindestens 3:1 gegenüber angrenzenden Farben.

Das betrifft Icons (z. Suchlupe, Papierkorb), Umrandungen von Formularfeldern, Buttons, Fokusindikatoren (z.B. sichtbarer Rahmen beim Tab-Fokus) sowie Diagramm-Linien, Legenden, und Statusanzeigen.

3. Navigation und Bedienung

- ☐ Kann man die Website vollständig mit der Tastatur bedienen (ohne Maus oder Touch)?

Ist alles mit Tab, Enter, Pfeiltasten erreichbar und nutzbar?

Typische Fehlerquellen:

- Buttons, die nur auf Mausklick reagieren
- Individuelle Komponenten ohne Tastaturfokus
- Tastaturfallen: Ein Benutzer muss mit der Tabulatortaste in ein Formularfeld gelangen und es mit Shift+Tab wieder verlassen können.

- ☐ Ist das Menü übersichtlich und logisch aufgebaut?

Erfolgskriterium: **Sinnvolle Reihenfolge**

☞ Wie's aussieht, so soll's auch vorgelesen werden.

Der Inhalt muss in einer sinnvollen Reihenfolge im Code stehen, sodass er auch beim Navigieren mit Tastatur oder Screenreader logisch verständlich bleibt.

Was heißt das konkret?

- Die visuelle Reihenfolge der Inhalte muss sich auch im Quellcode widerspiegeln.
- Keine Layout-Tricks (z.B. CSS-Positionierung oder Tabindex-Manipulation), die den Lesefluss verwirren.
- Klare Fokus-Reihenfolge: Der Nutzer kann einem sinnvollen Lesefluss folgen: Navigation → Hauptinhalt → Footer etc.

Prüfen:

- ☒ Navigiere mit der Tab-Taste oder einem Screenreader durch die Seite:
- Kommt der Text in der erwarteten Reihenfolge?
 - Sind Navigation, Inhalte, Formulare logisch aufgebaut?

Beispiele

- Zwei Spalten im Layout: Visuell links = zuerst im Code.
- In einem Formular folgt das Eingabefeld direkt nach der zugehörigen Beschriftung.
- Keine Sprünge im Fokus durch unlogisches tabindex.

- ☐ Gibt es einen sichtbaren Fokus beim Navigieren (z. B. Markierung bei Tab-Taste)?

Erfolgskriterium: **Sensorische Eigenschaften**

☞ Form, Farbe oder Position allein reichen nicht

– beschreib, was gemeint ist!

Was ist zu tun?

Vermeidung von Anweisungen, die sich ausschließlich auf sensorische Merkmale beziehen (z. B. „Klicken Sie auf den grünen Knopf“)

Erfolgskriterium: Seite mit Titel versehen

👁️ Yes, we do judge the page by its title.

Der Titel einer Seite beschreibt deren Inhalt eindeutig. Screenreader lesen den Titel als Erstes vor und er erscheint im Browsertab, in Lesezeichen und Suchergebnissen.

Prüfen:

Jede Seite muss einen aussagekräftigen, eindeutigen Titel im <title>-Element haben, der den Inhalt oder Zweck der Seite klar beschreibt.

➤ ☐ Bietet die Website verschiedene Methoden / mehrere Wege zur Navigation an?

🕒 Gib dem Nutzer mehr als einen Weg – nicht nur eine Tür.

Es muss mehr als einen Weg geben, um eine bestimmte Seite oder Funktion zu erreichen – z.B. über Hauptnavigation, Suche, Sitemap, Breadcrumbs, Kontextverlinkungen im Fließtext

4. Farben & Kontraste

➤ ☐ Ist der Text gut lesbar (z.B. keine weiße Schrift auf hellem Hintergrund)?

👁️ Wenn man's kaum lesen kann, ist es nicht barrierefrei.

Text (inkl. Text in Bildern) muss einen ausreichenden Kontrast zum Hintergrund haben:

- Normaler Text: mindestens 4,5:1
- Großer Text (ab 18pt oder 14pt fett): mindestens 3:1
- Ausnahmen: Logos oder rein dekorative Texte

Achte auf gute Lesbarkeit bei allen Lichtverhältnissen. Nutze CSS color und background-color mit hohem Kontrast (z.B. CSS: color: #000000; background-color: #FFFFFF;).

➤ ☐ Wichtige Infos sind nicht nur durch Farbe gekennzeichnet (z.B. rot = Fehler)?

Erfolgskriterium: Verwendung von Farbe

👁️ Farbe ist schön – aber nicht allein verständlich!

Informationen dürfen nicht ausschließlich durch Farbe vermittelt werden und es muss eine zusätzliche Kennzeichnung geben, z.B. durch Text, Symbole oder Muster.

Was heißt das konkret?

❌ „Felder mit rotem Rand sind Pflichtfelder“

✅ „Pflichtfelder sind rot markiert und mit * gekennzeichnet“

❌ „Klicke auf die grüne Schaltfläche, um fortzufahren“

✅ „Klicke auf die grüne Schaltfläche mit dem Text ‚Weiter‘“

Prüfen

- Werden Inhalte auch ohne Farbe verständlich?
- Simuliere Farbblindheit oder drucke in Graustufen.

5. Videos & Animationen

➤ ☐ Haben Videos Untertitel oder eine Textalternative?

🗣️ Wenn jemand spricht, muss man's auch lesen können.

➤ ☐ Gibt es bei automatischen Slideshows eine Pausenfunktion?

Erfolgskriterium: Reine Audio- und Videoinhalte (aufgezeichnet & live)

👁️ 🗣️ Ohne Bild braucht's Text, ohne Ton braucht's Beschreibung.

🔊 Live verstehen – Untertitel an!

Was ist zu tun?

Bereitstellung einer Textbeschreibung für ein aufgezeichnetes Video ohne Ton,
Bereitstellung einer Textalternative für reine Audio-Dateien (z.B. Podcast) sowie
Bereitstellung von Untertiteln für einen Live-Webcast

6. Formulare & Buchungstools

- ☐ Sind Formularfelder beschriftet (z. B. „Name“, „E-Mail“)?
 - 🗨 Ein Feld ohne Beschriftung ist wie eine Frage ohne Fragezeichen. Wenn Nutzer*innen Eingaben machen müssen, sollen sie klar erkennbare Beschriftungen oder Anleitungen erhalten, um die Eingabe korrekt vorzunehmen.
- ☐ Wird bei Fehlern (z. B. vergessene Angabe) klar erklärt, was fehlt?
 - 🕒 Ein Fehler ist nur halb so wild – wenn man ihn versteht.
 - Wenn ein Eingabefehler auftritt (z. Pflichtfeld vergessen, ungültige Eingabe), muss der Fehler klar beschrieben und identifiziert werden.
 - D.h. Nutzer*innen müssen verstehen, was falsch war – und wo
 - Fehlerhinweise müssen in Textform erfolgen (nicht nur visuell)

Typische Fehlerquellen

✗ Kein sichtbares Label oder nur Platzhaltertext

✗ Unklare Begriffe wie „Wert eingeben“

☑ „E-Mail-Adresse“ als Label + Hinweis: „Format: name@domain.at“

Technik-Tipp

🔗 Verwende <label for=“...”> und stelle eine programmatische Verknüpfung zum Eingabefeld her

7. Technik & Tests

- ☐ Lässt sich die Website auf mobilen Geräten gut bedienen – auch durch Assistenz-Systeme?

Erfolgskriterium: Name, Rolle, Wert

🗨 Name, Rolle, Wert – damit Technik weiß, was Sache ist.

Buttons haben programmatisch definierte Rollen und Namen z. B. <button aria-label=“Suche“>. Benutzeroberflächenelemente (z.B. Formulare, Buttons, Widgets) müssen eine programmatisch erkennbare Bezeichnung (Name), Rolle und ggf. einen aktuellen Wert haben, die für assistive Technologien verfügbar sind.

Was heißt das konkret?

➡ Screenreader & Co. müssen wissen:

- Was ist das? (z.B. Rolle = Button)
- Wie heißt es? (z.B. Name = „Suche“)
- Welcher Zustand gilt? (z. B. aktiviert/deaktiviert, ausgewählt)

- ☐ Besteht sie einen Basis-Test mit Tools wie:
www.wave.webaim.org oder <https://accessibilitychecker.org>

🔑 Wichtig:

Diese Checkliste ersetzt keine offizielle Prüfung, bietet aber eine gute erste Einschätzung. Für eine tiefere Analyse empfiehlt sich ein Test durch Webagenturen mit Barrierefreiheit-Expertise.