

Easy Web Check

Andreas Burkard & Gottfried Zimmermann

Kompetenzzentrum Digitale Barrierefreiheit, Hochschule der Medien, Stuttgart

Version 1 – Jan 2023

Hinweis: Der Easy Web Check ist eine vereinfachte Prüfmethode, mit der man in kurzer Zeit und mit kostenlosen Werkzeugen Barrieren in Webseiten finden kann. Dazu sind keine tiefen technischen Kenntnisse erforderlich. Achtung: Der Easy Web Check ist kein Ersatz für ausführlichere Prüfmethoden, wenn es um den Nachweis für Barrierefreiheit geht.

Anmerkung: Wir empfehlen vor der ersten Durchführung der Checkliste die benötigten Werkzeuge (siehe [Werkzeugliste](#) unten) zu installieren.

Siteimprove Accessibility Checker (Browser-Erweiterung)

- (1) Bilder haben jeweils einen Alternativtext (dieser darf auch leer sein, er darf nur nicht komplett fehlen).
- (2) (Bild-)Links oder grafische Bedienelemente haben jeweils entweder eine Beschriftung oder einen Alternativtext.
- (3) Alle Formularelemente haben programmatisch korrekt verknüpfte Beschriftungen.
- (4) Die Kontrastunterschiede von allen Texten im Vergleich zum Hintergrund sind mindestens 4,5:1 bei normaler Größe und mindestens 3:1 bei großer Schrift (ab 24 px bzw. 18,7 px bei fetter Schrift).
- (5) Die Sprache auf der Seite wurde korrekt per `lang`-Attribut im `<html>`-Element festgelegt.
- (6) Fließtextlinks sind ausreichend vom restlichen Fließtext zu unterscheiden (die Links sind entweder unterstrichen, fettgedruckt und/oder haben einen Kontrastunterschied zum restlichen Text von mindestens 3:1).
- (7) Die `id`-Attribute aller Elemente (falls sie welche haben) sind jeweils einzigartig.
- (8) Es sind keine leeren Überschriften vorhanden.
- (9) Es sind keine leeren Listen vorhanden.
- (10) Texte werden bei einem Browserzoom auf 200% nicht abgeschnitten und überlagern nicht.
- (11) Die sichtbare Beschriftung von Links und/oder Buttons und deren Alternativtexte stimmen überein (im Alternativtext muss mindestens alles auch vorkommen, was visuell als Beschriftung zu lesen ist).

„Inhalte gegliedert“ Bookmarklet

- (12) Es ist nur *eine* `<h1>`-Überschrift vorhanden, die den Inhalt der Seite zusammenfasst.
- (13) Die Überschriftenstruktur ist sinnvoll (z. B.: wenn eine Überschrift einer anderen untergeordnet ist, sollte sie eine höhere Überschriftenebene haben).
- (14) Es werden nicht grundlos Überschriftenebenen übersprungen.
- (15) Überschriften, die visuell nach Überschriften aussehen und vom Sinn her welche sind, sind als `<h1>` - `<h6>` HTML-Überschriften implementiert.
- (16) Texte, die keine Überschriften sein sollen, sind auch **nicht** als `<h1>` - `<h6>` HTML-Überschriften implementiert (beispielspielweise um visuell einen Text hervorzuheben).
- (17) Es werden überall die Tags `` und `` statt den veralteten Tags `` und `<i>` verwendet, um Texte fett oder kursiv zu machen.
- (18) Textabsätze werden in `<p>`-Elementen untergebracht.
- (19) Es werden keine `<p>`-Elemente in Kombination mit Leerzeichen verwendet, um Leerzeilen zu erzeugen.

„Lists Bookmarklet“

- (20) Zusammengehörige Bedienelemente werden korrekt geschachtelt in einer HTML-Liste zusammengruppiert.
- (21) Zusammengehörige Navigationselemente werden korrekt geschachtelt in einer HTML-Liste zusammengruppiert.
- (22) Linklisten werden in einer HTML-Liste zusammengruppiert.
- (23) Kacheln werden in einer HTML-Liste zusammengruppiert.
- (24) Die Brotkrümelnavigation wird in einer HTML-Liste zusammengruppiert.

„Tables“ Bookmarklet

- (25) Die Spalten- und Zeilenüberschriften sind korrekt mit `<th>`-Elementen ausgezeichnet.
- (26) Jede Tabelle, die visuell wie eine Tabelle aussieht und vom Sinn her eine ist, wird als HTML-Liste ausgezeichnet (und wird durch das Bookmarklet markiert).

taba11y

- (27) Alle Bedienelemente sind mit der Tastatur fokussierbar (haben in „taba11y“ eine Markierung und Nummer).
- (28) Die Reihenfolge der Tab-Navigation (mit der Tastatur) ist sinnvoll.

„Images“ Bookmarklet

- (29) Bilder (die keine Bedienelemente sind) und wichtige Informationen enthalten, haben sinnvolle Alternativtexte, welche die Informationen äquivalent wiedergeben.
- (30) Bilder (die keine Bedienelemente sind), die nur dekorativ sind, haben leere Alternativtexte `alt=""`.
- (31) Grafische Bedienelemente / Bildlinks beschreiben ihre Funktion (z. B.: „Filter“) oder ihr Ziel (z. B.: „Twitter besuchen“).
- (32) Alternativtexte sind nicht redundant zum umliegenden Text.
- (33) Klickbare Logos sind beschrieben und nennen das Ziel, z. B.: „Logo der HdM – zur Startseite“.
- (34) Es werden keine Schriftgrafiken verwendet (die Ausnahmen sind Logos und Fotos).

„Lang“ Bookmarklet

- (35) Textabschnitte, die nicht der Hauptsprache der Webseite entsprechen, müssen per `lang`-Attribut mit der entsprechenden Fremdsprache ausgezeichnet werden, z.B.: `lang="en"` für Englisch.¹

Web Developer

- (36) Wenn auf 200% vergrößert wird (bei einer Auflösung von 1280x768), darf das Layout nicht „zerbrechen“ und keine Funktionalität verloren gehen (Überprüfung: Im „Web Developer“: „Resize“ → „Resize Window...“ → 1280x768 eingeben, dann im Browser auf 200% zoomen).
- (37) Bei einer Bildschirmgröße von 334px brechen die Inhalte der Seite responsiv um, sodass kein horizontales Scrollen erforderlich ist (Überprüfung: Im „Web Developer“: „Resize“ → „Resize Window...“, bei der Breite 334px eingeben). Ausnahmen: Elemente mit zweidimensionalen Layouts wie z.B. Bilder, Videos, Landkarten, Diagramme, Spiele, Datentabellen und Applikationen mit permanent verfügbaren Werkzeugleisten).

WCAG Color Contrast Checker

- (38) Die Kontraste von allen Texten im Vergleich zum Hintergrund sind mindestens 4,5:1 bei normaler Größe und mindestens 3:1 bei großer Schrift (ab 24 px bzw. 18,7 px bei fetter Schrift).

¹ W3Schools - Liste der Ländercodes: https://www.w3schools.com/tags/ref_language_codes.asp

Colour Contrast Analyser

- (39) Bedienelemente haben einen Kontrastunterschied von mindestens 3:1 zum Hintergrund (inklusive ihrer verschiedenen Zustände).

"Textabstände" Bookmarklet

- (40) Nach Einsatz des Bookmarklets "Textabstände" sollte das Layout nicht „zerbrechen“.

Landmarks

- (41) Wenn mehrere Bereichs-Elemente und/oder ARIA-Landmarks des gleichen Typs (z. B. „Navigation“) verwendet werden, dann benötigen diese per `aria-label` Attribut einen einzigartigen Namen, sonst ist das verwirrend bei der Navigation z. B. für Screenreader-Nutzende.

Autocomplete Attribute Favlet

- (42) Jedes Formularfeld, in das Nutzerbezogene Daten eingetragen werden sollen, müssen als Attribut `autocomplete` mit der passenden Belegung enthalten².

Tastaturnavigation (manuell)

- (43) Jedes Element, das mit der Maus bedienbar ist, muss auch mit der Tastatur bedienbar sein.
- (44) Mit der Tab-Taste kann man auf jedes Bedienelement navigieren und der Tastaturfokus ist visuell deutlich sichtbar (z. B. als Rahmen oder als Farbänderung, die zum Normalzustand einen höheren Kontrast als 3:1 hat).
- (45) Es ist keine Tastaturfalle vorhanden.
- (46) Man kann nicht hinter Dialogfeldern, Cookie-Warnungen und Menüs (z. B. dem mobilen Navigationsmenü) im visuell unsichtbaren Bereich navigieren.
- (47) Wenn man ein Menü ausklappt, ist der erste Menüpunkt das nächste Element, auf das man mit der Tab-Taste oder mit den Pfeiltasten navigieren kann.

Sichtprüfung

- (48) Diagramme oder Abbildungen sind auch ohne Farbwahrnehmung sinnvoll lesbar.
- (49) Der Titel des Tabs (Dokumenttitel) enthält sowohl den Namen des Webauftritts als auch den der Unterseite, auf der man sich befindet.
- (50) Linktexte sind aussagekräftig (und nicht nur URLs, außerdem sollten Download-Links das Format des Downloads angeben).

² WCAG 2.1, Abschnitt 7 Input Purposes for User Interface Components: <https://www.w3.org/TR/WCAG21/#input-purposes>

- (51) Die Texte von Überschriften sind aussagekräftig, sodass klar ist, was im folgenden Abschnitt zu erwarten ist.
- (52) Die Beschriftungen von Formularelementen sind aussagekräftig, sodass klar ist, was für eine Eingabe im Formularelement erwartet wird.
- (53) Es gibt mindestens zwei verschiedene Möglichkeiten, Webseiten zu erreichen (z. B. Navigationsmenü, vollständige Verlinkungen, Sitemap, Suche).
- (54) Die Navigation ist über die Unterseiten eines Webauftritts hinweg konsistent.
- (55) Bei Sounds, die beim Laden der Website automatisch starten und länger als 3 Sekunden dauern, muss es eine Möglichkeit geben, diese abzuschalten.
- (56) Bewegte Inhalte (z. B. animierte GIFs), die nicht nach 5 Sekunden automatisch stoppen, müssen abschaltbar sein.
- (57) Die Seite darf keine Inhalte enthalten, die häufiger als 3-mal pro Sekunde flackern.
- (58) Formularelemente müssen eine sichtbare und sinnvolle Beschreibung haben.
- (59) Wenn es eine Zeitbegrenzung gibt (z. B. bei Online-Transaktionen wie die Session einer Bank, zeitbegrenzte Dialoge oder das automatische Neu-Laden einer Webseite) muss die Zeitbegrenzung angezeigt werden und man muss das Zeitlimit verlängern oder ausschalten können.
- (60) Wenn die Anmeldung über biometrische Daten erfolgt, gibt es dazu eine Alternative.
- (61) Das Ändern der Bildschirmausrichtung bei mobilen Geräten darf nicht verhindert werden, außer es ist essenziell für die Anwendung.
- (62) Folgende Elemente sind im Webauftritt vorhanden:
 - a. Erläuterung der Website in Deutscher Gebärdensprache
 - b. Erläuterung der Website in Leichter Sprache
 - c. Erklärung zur Barrierefreiheit

Zeiger-, Bewegungs- und Gestensteuerung (manuell)

- (63) Wenn eine komplexe Zeigergeste (z. B. Wischgesten oder Gesten mit mehreren Fingern) bei einem Bedienelement verwendet werden kann, muss es auch eine Alternative mit einer einfachen Zeigergeste geben.
- (64) Ein Bedienelement darf erst ausgeführt werden, wenn die Maustaste / die Eingabe mit einem Eingabegerät oder Finger losgelassen wird, nicht direkt beim Drücken.
- (65) Wenn Webanwendungen auf Bewegungssteuerung reagieren (beispielsweise über eine Kamera oder über die Bewegungen eines Mobilgerätes), muss es eine barrierefreie Alternative geben.

Videoplayer

- (66) Verwendete Videoplayer müssen in der Lage sein Untertitel und/oder Audiodeskriptionen ein- und ausschalten können (je nachdem, was vorhanden ist)
- (67) Die Bedienelemente von verwendeten Videoplayern für Untertitel und/oder Audiodeskriptionen (je nachdem, was vorhanden ist) sollten sich auf der gleichen Ebene befinden wie die Start- und Stopp-Buttons befinden.
- (68) Verwendete Videoplayer geben Untertitel und/oder Audiodeskriptionen (je nachdem, was vorhanden ist) synchron zum Gesprochenen/Visuellem aus.
- (69) Benutzende können die Darstellung der Untertitel an ihre Bedürfnisse anpassen (z.B. Größe, Schriftart, Farbe; Transparenz, Farbe und Rand für Hintergrundbox).

Video- und Audiodateien

- (70) Jede Audiodatei, in der gesprochen wird, hat ein Transkript. Ein Transkript ist ein Text, der das Gesprochene, wer spricht, relevante Geräusche, Musik und visuelle Informationen (nur bei einem Video) in Textform enthält.
- (71) Jedes stumme Video hat eine nicht visuelle Alternative (entweder Transkript oder Audiodatei).
- (72) Videos mit Ton haben Untertitel, wenn darin gesprochen wird oder es Geräusche und/oder Musik gibt, die inhaltlich relevant sind.
- (73) Videos haben eine Audiodeskription, wenn man ohne die visuellen Informationen des Videos wichtige Inhalte verpassen würde. Bei der Audiodeskription wird das visuell Sichtbare per Audio beschrieben (entweder in einer alternativen Tonspur oder einer alternativen Version des Videos).

Fehlermeldungen

- (74) Bei fehlerhaften Eingaben muss eine gut verständliche Fehlermeldung erscheinen.
- (75) Die fehlerhaft ausgefüllten Formularfelder müssen identifizierbar und in Textform beschrieben werden (nicht nur visuell markiert).
- (76) Es sollte das Format angegeben werden, in dem die Daten eingetragen werden, beispielsweise das Datumsformat „(DD.MM.YYYY)“.

Dokumentation und technischer Support

- (77) Dokumentation (falls vorhanden; Erklärung zur Barrierefreiheit gilt auch als Dokumentation), ist selbst barrierefrei bzw. erfüllt die (anderen) Prüfschritte dieser Checkliste.
- (78) Ein technischer Support muss über mindestens zwei Kommunikationswege erreichbar sein (z.B. E-Mail, Chat, Telefon und/oder Videotelefonie).

- (79) Wenn Dokumentation vom technischen Support als Webseiten bereitgestellt werden, müssen diese selbst barrierefrei sein, d.h. die (anderen) Prüfschritte dieser Checkliste erfüllen.

Werkzeugliste

- "Siteimprove Accessibility Checker" Browser Erweiterung ([Google Chrome](#)³ | [Firefox](#)⁴)
- Bookmarklet "Inhalte gegliedert" aus der [BIK BITV Werkzeugliste Bookmarklets](#)⁵
- "Lists Bookmarklet" aus der [Paul J. Adam's Bookmarklet-Liste](#)⁶
- "Tables Bookmarklet" aus der [Paul J. Adam's Bookmarklet-Liste](#)⁷
- "taba11y" ([für Google Chrome](#)⁸)
- "Images Bookmarklet" aus [Paul J. Adam's Bookmarklet-Liste](#)⁹
- "Lang Bookmarklet" aus der Tabelle in der [Paul J. Adam's Bookmarklet-Liste](#)¹⁰
- Browser Erweiterung "Web Developer" ([Google Chrome](#)¹¹ | [Firefox](#)¹²)
- WCAG Color contrast checker ([Google Chrome](#)¹³ | [Firefox](#)¹⁴)
- [Colour Contrast Analyser](#)¹⁵
- Bookmarklet "Textabstände" aus der [BIK BITV-Test Werkzeugliste Bookmarklets](#)¹⁶
- Browser Erweiterung "Landmarks" ([Google Chrome](#)¹⁷ | [Firefox](#)¹⁸)

³ Siteimprove Accessibility Checker. Google Chrome: <https://chrome.google.com/webstore/detail/siteimprove-accessibility/djcglbmbegflehmfbfleeckjihmedcopn>.

⁴ Siteimprove Accessibility Checker. Firefox: <https://addons.mozilla.org/de/firefox/addon/si-accessibility-checker/>

⁵ „Inhalte gegliedert“ Bookmarklet (BIK BITV-Test Werkzeugliste):

https://www.bitvtest.de/bitv_test/das_testverfahren_im_detail/werkzeugliste/bookmarklets.html

⁶ „Lists Bookmarklet“ (Paul J. Adam's Bookmarklet-Liste): <https://pauljadam.com/bookmarklets/lists.html>

⁷ „Tables Bookmarklet“ (Paul J. Adam's Bookmarklet-Liste): <https://pauljadam.com/bookmarklets/tables.html>

⁸ taba11y. Google Chrome: <https://chrome.google.com/webstore/detail/taba11y/aocppmckdocdkphmofnklcjhdidmgga>

⁹ „Images Bookmarklet“ (Paul J. Adam's Bookmarklet-Liste): <https://pauljadam.com/bookmarklets/images.html>

¹⁰ „Lang Bookmarklet“ (Paul J. Adam's Bookmarklet-Liste): <https://pauljadam.com/bookmarklets.html>

¹¹ Web Developer. Google Chrome: <https://chrome.google.com/webstore/detail/web-developer/bfbameneiokkgbdmiekhjnmfkcnldhnm?hl=de>

¹² Web Developer. Firefox: <https://addons.mozilla.org/de/firefox/addon/web-developer/>

¹³ WCAG Color contrast Checker. Google Chrome: <https://chrome.google.com/webstore/detail/wcag-color-contrast-check/plnahcmalebfmaghcpcmpaciebdhgdf?hl=en>

¹⁴ WCAG Contrast Checker. Firefox: <https://addons.mozilla.org/de/firefox/addon/wcag-contrast-checker/>

¹⁵ Colour Contrast Analyser (CCA): <https://www.tpgi.com/color-contrast-checker/>

¹⁶ Bookmarklet „Textabstände“ (BIK BITV-Test Werkzeugliste):

https://www.bitvtest.de/bitv_test/das_testverfahren_im_detail/werkzeugliste/bookmarklets.html

¹⁷ Landmarks. Google Chrome: <https://chrome.google.com/webstore/detail/landmark-navigation-via-k/ddpokpbjopmeeiiolheejipkonlklgp>

¹⁸ Landmarks. Firefox: <https://addons.mozilla.org/de/firefox/addon/landmarks/>

- "Autocomplete Attribute Favlet" Bookmarklet von [Level Access Web Labs](#)¹⁹

Hinweis

Dieses Dokument ist unter der Lizenz [CC-BY 4.0](#) freigegeben.

Bitte senden Sie Ihr Feedback zu dieser Checkliste und zu Easy Web Check [an das Kompetenzzentrum Digitale Barrierefreiheit](#).

Wenn Sie mehr über den Easy Web Check erfahren möchten oder sich und Ihre Mitarbeitenden darin schulen lassen möchten, empfehlen wir Ihnen unseren [Mini-Workshop „Easy Web Check – Webseiten einfach auf Barrierefreiheit testen“](#), der regelmäßig online stattfindet.

¹⁹ "Autocomplete Attribute Favlet" (Level Access Web Labs):
https://labs.levelaccess.com/index.php/Autocomplete_Attribute_Favlet