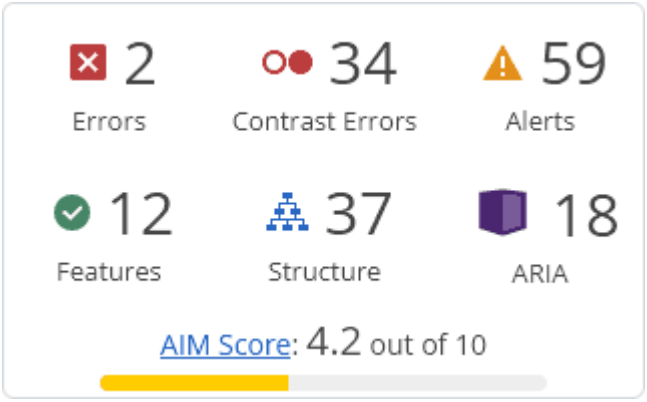
Kreuzbund Website

Bericht und Leitfaden „Barrierefreiheit“

Im Oktober 2026 wurden allgemeine Maßnahmen zur Stärkung der Barrierfreiheit der Website www.kreuzbund.de durchgeführt. Hierzu wurden insbesondere Standard-Tests nach WAVE/WebAIM durchgeführt und die sich daraus ergebenden Handlungsempfehlungen umgesetzt. In diesem Leitfaden werden die durchgeführten Maßnahmen so beschrieben, dass sie von Administratoren anderer Kreuzbund-Websites, welche auf dem Template der Bundesgeschäftsstelle beruhen, nachvollzogen werden können.

Stand der Dinge

Ein erster Test der Startseite ergibt ein solches Bild:



Vorgehensweise:
Zu den aufgeführten kritischen Hinweisen (Errors, Contrast Errors) und den Alerts lässt man sich die einzelnen Quellen/Fundorte anzeigen und behebt diese – wo immer möglich – Schritt für Schritt. Dabei sind die wesentlichen Punkte, an die man herangeht, die konkreten Fehler (Errors = Strukturelle Fehler und Contrast Errors = Kontrastfehler) und – je nach Einschätzung – die Meldungen im Bereich „Warnungen“ („Alerts“).

I. FEHLER

Test Startseite:

Ergebnis:
- **2 Fehler**

Die Fehler inkl. der HTML-Codes (Fehler hervorgehoben):

1. **Empty buttonReference:**
   ```
   <input id="oneo-nav-search-112" type="search" name="keywords">
   <button type="submit"></button>
   ```

2. **Empty linkReference:**
   ```
   <div class="ce_rsce_oneo_nav_search main-navigation-mobile-search">
   <a href=""></a>
   ```

Lösung:
Beide Fehler lassen sich in den Contao-Einstellungen korrigieren; sie betreffen das Modul „Navigationssuche“, von dem 2 Instanzen in der Installation laufen:

- Main Navigation Search
- Mobile Navigation Search

Für beide Module gilt:

1. Angabe im Feld „Such-Seite“ setzen
2. Werte in „Suchfeld-Bezeichnung“ und „Button-Bezeichnung“ eintragen

→ Fertig. Damit sind die beiden Fehler behoben, die Fehler-Anzeige sinkt auf 0.

Andere Seiten wiesen anschließend keine weiteren Fehler auf.

II. Kontrast-FEHLER

ℹ Allgemeiner Hinweis zum Kontrast: Angestrebt wird ein Kontrastverhältnis von mindestens 4,5:1 (genügt dem Standard WCAG Level AA).

Ergebnis:
- **34 Kontrast-Fehler**

Beschreibung:
Alle 34 Kontrastfehler werden als „Very low contrast“ deklariert. Wichtig: Diese Kontrast-Schwäche bezieht sich auch auf den Kontrast Weiß/Kreuzbund-Grün! Also eine kreuzbund-grüne Schrift auf einem weißen Hintergrund wird als zu schwacher Kontrast für den WCAG Lvl. AA-Test ermittelt. Das Kontrastverhältnis zwischen #fff (weiß) und #009d8b (kreuzbund-grün) liegt nämlich nur bei 3,7:1. Dieser Fall betrifft auch die meisten der gefundenen Kontrast-Fehler.

Lösung:
Ich wähle statt des Original-Kreuzbund-Grüns die Ersatzfarbe #008577.
Die weist zu #fff (weiß) einen Kontrast von 4,45:1 auf, und erfüllt damit die WCAG Level AA-Vorgaben.

(Um die strengere „WCAG Level AAA“-Vorgabe zu erfüllen, müsste ich den Kontrast weiter steigern auf mind. 7:1 und würde dann z.B. bei #01635f rauskommen als Ersatz für das Kreuzbund-Grün; damit bin ich dann aber so weit von der Original-Farbe entfernt, dass ich darauf verzichte.)

Die Lösungen im Einzelnen:

1. Element: „Hero Titeltext“

Für alle Elemente, bei denen Schrift auf einem Hintergrundbild stattfindet, wird jetzt eine explizierte Hintergrundfarbe angegeben, die mit der Schriftfarbe ausreichend kontrastiert. Damit das beim Hero-Image funktioniert, gehe ich so vor:

Kreuzbund-CSS (files/kreuzbund/css/style.css):
Neu einfügen (am besten beim .hero-Block):

```
.hero .mod_randomImage { z-index: 1; }
.hero .centered-wrapper-inner { z-index: 2 }
```

(durch diese Anweisungen wird verhindert, dass beim Setzen einer Hintergrundfarbe diese alles verdeckt und Bild und Schrift nicht mehr sichtbar wären.)

Contao:
• Element "Zentrierter Bereich Anfang" im Hero-Artikel:
    • Eigene Hintergrundfarbe: 008577

→ Fertig.

2. Element: Lauftext "Wir sind zurzeit dabei"

Hier besteht das gleiche Problem mit dem weißen Text vor dem Hintergrundbild. Wir weisen der Sektion einfach eine kontrastreiche Hintergrundfarbe zu:

Contao:
• Element "Zentrierter Bereich Anfang" im Laufschrift-Artikel:
    • Eigene Hintergrundfarbe: 008577

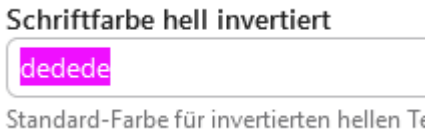
→ Fertig.

3. Element: Textfarbe "Selbsthilfegruppen*" und "Menschen" im Bereich "Zahlen und Fakten")

Das Problem ist diese CSS-Anweisung:

```
.-color-inverted .counter-boxes-item h3 { #dedede; }
```

Dieses Problem könnten wir dadurch lösen, dass wir in der Kreuzbund-CSS eine neutralisierende Anweisung hinterlegen. Besser lässt sich das Problem aber direkt global lösen, indem wir die ONEO-Farbcodes ändern. Dazu nutzen wir das Modul „Rocksolid Theme Assistant“ (bei Bedarf nachzuinstallieren) und setzen im Bereich „Schriftfarbe hell invertiert“ den Wert von dedede auf ffffff:



→ Fertig.

4. Allgemeine Farb-Einstellungen

Bei der Gelegenheit korrigiere ich direkt alle Farben, die zentral im Rocksolid-Theme-Assistant korrigiert werden können. Das betrifft alle Felder im Bereich „Schrift- und Elementfarben“ und „Links und Navigation“ mit den Einträgen „009688“ (ich nehme an, das war eine ältere Iteration des Kreuzbund-Farbtons).

Die betreffenden Felder sind:

- (Absatz „Schrift- und Element-Farben“):
  - Auswahlfarbe
  - Überschriftfarbe 1
  - Überschriftfarbe 2
  - Hightlightfarbe
  - Confirmation-Farbe
- (Absatz „Links & Navigation“):
  - Link-Farbe
  - Navigationsfarbe Hover
  - Navigationsfarbe Aktiv
  - Dropdown Hover-Schriftfarbe

Ergebnis:
→ Startseite reduziert auf 1 Kontrast-Fehler i.d. Laufschrift (der ist nicht korrigierbar, da sich die Meldung auf ein unsichtbares, nicht änderbares Animations-Element der Laufschrift bezieht. Das Problem könnte nur durch einen Verzicht auf die Animation gelöst werden und wäre eine reine Formalität.)

→ Einfache Seiten ohne spezielle Effekte (z.B. "Impressum" usw.) haben nun null Kontrastfehler.

5. Weitere Seiten

Da sich der WAVE/AIM-Test natürlich immer nur auf die einzelne, getestete Seite bezieht, geht ich die einzelnen Seiten des Webauftritts Stück für Stück durch und korrigiere noch die letzten Fehler, die auf dem Weg erscheinen.

Diese sind:

• Unterseiten unter „Für wen wir da sind“

Fehler:
4 Kontrast-Fehler i.d. Laufschrift

Lösung:
Contao-Einstellungen:

• „Zentrierter Bereich Anfang“: Wert „Eigene Hintergrundfarbe“ auf 008577

Ergebnis:
→ reduziert auf 2 Fehler i.d. Laufschrift (nicht korrigierbar, da unsichtbare Animations-Elemente, vgl. o.)

• Seiten unter „Chat“ mit Bilder-Slider

Fehler:
Kontrastfehler Hintergrundfarbe "Slider"/Schrift
Lösung:
Kreuzbund-CSS (files/kreuzbund/css/style.css):

Ändern:

```
alt: .mod_rocksolid_slider.chatzitate .content-text { background: #aaa;
neu: .mod_rocksolid_slider.chatzitate .content-text { background: #656565;
```

Ergebnis:
→ 0 Fehler

• Seite „Sitemap“

Fehler:
17 Kontrastfehler (betrifft die Zwischen-Überschriften)

Lösung:
Kreuzbund-CSS (files/kreuzbund/css/style.css):

```
alt: .mod_sitemap ul.level_2 li.submenu a.submenu { color: #777; }
neu: .mod_sitemap ul.level_2 li.submenu a.submenu { color: #333; }

alt: .mod_sitemap ul.level_3 li.submenu a.submenu { color: #aaa; }
neu: .mod_sitemap ul.level_3 li.submenu a.submenu { color: #656565; }
```

Ergebnis:
→ 0 Fehler

6. Allgemeine Elemente

Hier eine Liste weiterer, allgemeiner Inhaltselemente, die mir immer wieder begegnet sind und auf diversen betreffenden Seite zu ändern waren.

Zentrierter Bereich mit Hintergrundbild und weißer Schrift darauf
→ Beachten, dass im Element "Zentrierter Bereich (Anfang)" eine kontraststarke Hintegrundfarbe eingesetzt wird (z.B. die hier verwendete Grün 008577, oder ein Grau, das mindestens 767676 oder dunkler ist (z.B. wie häufig hier: 656565)

Button mit CSS-Klasse "grey" (z.B. in den Chat-Seiten)

→ Das mittelhelle Grau wird durch ein dunkleres ersetzt:

Lösung:
Kreuzbund-CSS (files/kreuzbund/css/style.css):

```
alt:
div.button-box.grey a { background: #aaa; }
div.button-box.grey a:hover, div.button-box.grey a:focus { background:
#777; }

neu:
div.button-box.grey a { background: #656565; }
div.button-box.grey a:is(:hover:focus) { background: #333; }
```

Button mit CSS-Klasse "-secondary" (z.B. auf der Seite „Das Kreuzbund-Forum“)

Wird bei einem Inhaltselement „Button“ die „Button-Variante: Button 2“ ausgewählt, so wird dem Button ebenfalls eine Hintergrundfarbe zugewiesen, die zu kontrastarm gegen weiß ist.

Lösung:
Kreuzbund-CSS (files/kreuzbund/css/style.css):

```
alt:
button.-secondary, .button.-secondary, input[type="submit"].-secondary,
.feature-box-link.-secondary, .headline-box-link.-secondary, .pricing-
table-plan a.-secondary { background: #aaa; border-color: #aaa; color: #fff
}
button.-secondary:is(:hover, :focus), .button.-secondary:is(:hover,
:focus), input[type="submit"].-secondary:is(:hover, :focus), .feature-box-
link.-secondary:is(:hover, :focus), .headline-box-link.-
secondary:is(:hover, :focus), .pricing-table-plan a.-secondary:is(:hover,
:focus) { background: #555; border-color: #777 !important }

neu:
button.-secondary, .button.-secondary, input[type="submit"].-secondary,
.feature-box-link.-secondary, .headline-box-link.-secondary, .pricing-
table-plan a.-secondary { background: #656565; border-color: #656565;
color: #fff }
button.-secondary:is(:hover, :focus), .button.-secondary:is(:hover,
:focus), input[type="submit"].-secondary:is(:hover, :focus), .feature-box-
link.-secondary:is(:hover, :focus), .headline-box-link.-
```

```
secondary:is(:hover, :focus), .pricing-table-plan a.-secondary:is(:hover,
:focus) { background: #333; border-color: #333 !important }
```

Element "Downloads": Anzeige d. Dateigröße

→ Die Farben im Element werden dunkler gemacht; zusätzlich wird das Element „Dateigröße“ etwas verkleinert, um sich wieder besser abzuheben.

Lösung:
Kreuzbund-CSS (files/kreuzbund/css/style.css):

```
alt:
.content-downloads a { color: #333; ...
.content-downloads span.size { color: #aaa; }

neu:
.content-downloads a { color: #000; ...
.content-downloads span.size { color: #656565; font-size: .8rem }
```

Element "Boxen (Team")

→ Kontrastfehler für Kontakt-Icons (z.B. E-Mail-Icon), die bei Hover/Mouseover erscheinen.

Lösung:
Oneo-Template (/templates/oneo/rsce_oneo_team_boxes.html5):

```
alt:
'email' => array(
      'color' => '888888',

neu:
'email' => array(
      'color' => '656565',
```

ℹ Ggf. müssen noch weitere Farben angepasst werden, falls andere Kontaktmöglichkeiten verwendet werden und der WAVE/AIM-Test einen entsprechenden Kontrastfehler dafür anzeigt)

CSS-Alternative:
Falls Sie die Anpassung lieber im CSS vornehmen, ergänzen Sie dort die Zeile:

```
.team-boxes-item .team-boxes-social-links > li > a { color: #656565
!important }
```

Timeline (hier: Seite "Geschichte des Kreuzbunds")

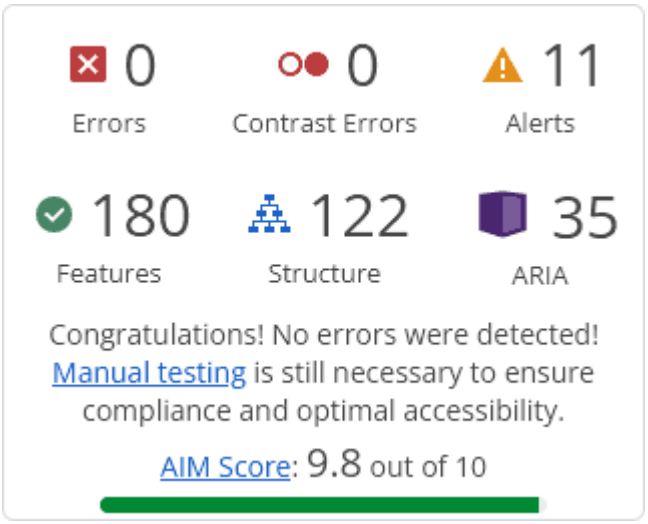
→ Kontrastfehler für die Überschriften "Jahreszahl" und für Links im Text (d.h. Links im Element-Feld „Beschreibung“)

Lösung:
Kreuzbund-CSS (files/kreuzbund/css/style.css):

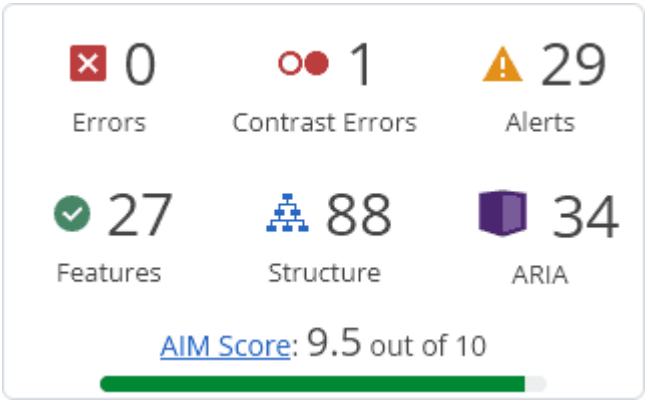
```
alt:
.timeline-item > h3, .timeline-item-text h3 { color: #9eb2bb; }

neu:
.timeline-item > h3, .timeline-item-text h3, .timeline-item-text a
 { color: #007c6f; }
```

Gesamtergebnis

Mit diesen Maßnahmen konnten alle Seiten auf 0 Kontrastfehler gebracht werden:



Nur auf den Seiten mit animierten Text-Effekten bleibt pro Laufschrift-Element ein Kontrastfehler bestehen.



III. Warnungen

Die Warnungen im WAVE/AIM-Test beziehen sich im Wesentlichen auf strukturelle Probleme – zum Beispiel eine falsche Reihenfolge von Überschriften oder Redundanzen einzelner Elemente.

Die auftretenden Warnmeldungen sind hier alphabetisch nach erstem Stichwort sortiert:

Warnung: Geräteabhängiges Script (Device dependent event handler)

Problem: Es gibt eigentlich kein Problem. Der WAVE/AIM-Test weist darauf hin, dass ein Script-Element gefunden wurde, das möglicherweise nicht auf allen Geräten funktionieren könnte.

Im Falle der kreuzbund.de-Website betrifft diese Meldung im wesentlichen die News-Listen-Seiten. Hier wird ein Script eingesetzt, das bewirkt, dass der User irgendwo auf einen News-Eintrag klicken kann, um zur Detail-Ansicht der News zu gelangen. Es braucht also nicht gezielt auf den Link „Weiterlesen…“ geklickt zu werden, sondern ein Klick so grob in die Richtung des Teasers reicht schon.

Dies geschieht über den Code

```
onclick="window.location.href='<?= $this->link ?>';"
```

, der im entsprechenden Newslist-Template verwendet wird.

Lösung: Es muss natürlich darauf geachtet werden, dass auch unabhängig von dieser Script-Lösung eine Möglichkeit besteht, zum Detail-Eintrag der Newsmeldung zu gelangen. Das ist hier in jedem Fall gegeben, z.B. über den „Weiterlesen…“-Link.

Aternativ: Möchte man den Fehler formal aus seinem WAVE/AIM-Score weg haben, könnte der o.a. Code entfernt werden, oder man umgibt den kompletten Teaser-Eintrag mit einem entsprechenden <a>-Tag.

Warnung: Link zu PDF-Dokument (Link to PDF document)

Problem:
Es gibt eigentlich kein Problem: Der WAVE/AIM-Test weist nur darauf hin, dass PDF-Dokumente ebenfalls barrierearm gestaltet werden sollten; der Test überprüft aber die konkreten verlinkten Dokumente nicht, sondern gibt

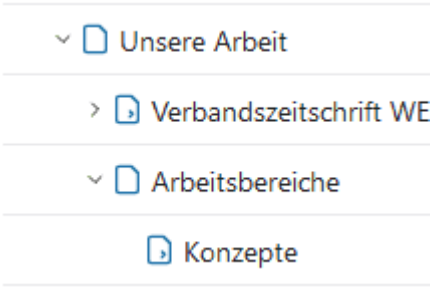
grundsätzlich bei verlinkten PDF-Dokumenten einen entsprechenden Hinweis aus.

Warnung: Redundanter Link (Redundant link)

Problem:
Der Test weist darauf hin, dass benachbarte Hyperlinks dasselbe Ziel haben.

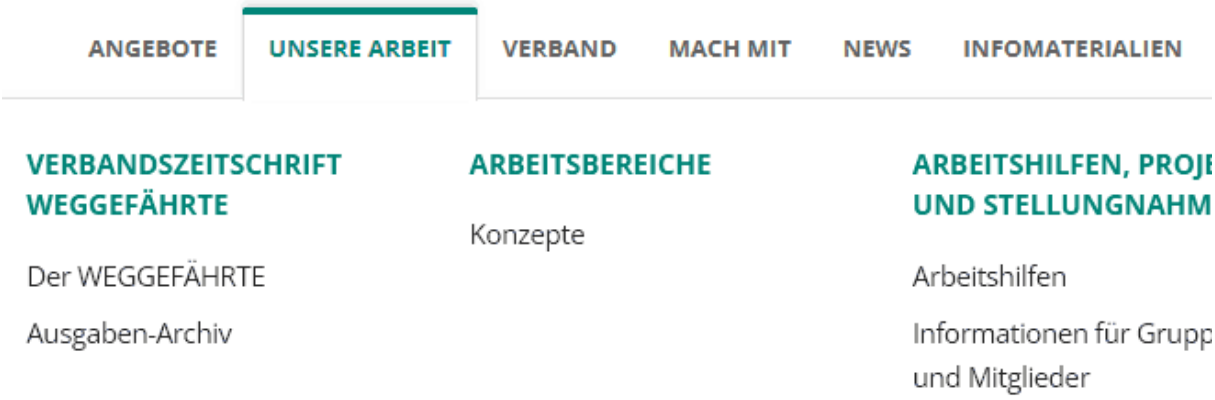
Das Phänomen tritt hier auf, weil die Website derzeit in den „Mega-Menüs“ (zu verwalten im Bereich „Layout > RockSolid Mega Menu“) teilw. Überschriften enthält, die jeweils nur eine verlinkte Seite enthalten. In diesen Fällen ist die Überschrift mit der identischen Seite verlinkt (bzw. als „Interne Weiterleitung“ angelegt). Das spiegelt sich auch in der Seiten-Struktur wieder:

Seiten-Struktur:



Die Seite „Arbeitsbereiche“ enthält nur eine Unterseite und selbst keine Inhalte.

Entsprechende Darstellung im Mega-Menü:



Die Überschrift „Arbeitsbereiche“ enthält nur eine Seite und da sie selbst keine Inhalte enthält, leitet sie auf die Seite „Konzepte“ weiter, was vom WAVE/AIM-Test (korrekterweise) als „identischer Link benachbarter Elemente“ deklariert wird.

Lösung:
Es gibt keine echte, befriedigende Lösung für dieses Problem. Man könnte im o.a. Beispiel eine Seite „Arbeitsbereiche“ schaffen, die nur einen Link zur Seite „Konzepte“ enthält; dadurch würde der Fehler verschwinden. Wenn man diesen Score im WAVE/AIM-Test erreichen möchte, müsste man dementsprechende Umbau-Arbeiten an der Seiten-Struktur durchführen, so dass keine redundanten Links durch Weiterleitungen mehr entstehen.

Warnung: Redundanter Titel-Text (Redundant title text)

Problem:
Das Contao-Navigationsmodul legt in der HTML-Ausgabe zu jedem Navigationslink im Hauptmenü ein „title“-Attribut an, welches gleichlautend ist mit dem Namen der Seite, auf die verlinkt wird (solange die Seite in den Seiten-Einstellungen keinen anderslautenden Titel, was aber standardmäßig nicht der Fall ist und auch nur selten anders gehandhabt wird).

Lösung:
Ich entferne das „title“-Attribut im Navigations-Template; selbst für den seltenen Fall, dass eine Seite einen anderen Titel als Namen hat, ergibt diese Darstellung im Navigationsmenü aus meiner Sicht keinen echten Nutzen.

Template (im Verz. /templates/oneo/):
nav_rsmm.html5:
Code:
```
title="<?= $item['pageTitle'] ?: $item['title'] ?>"
```
ersatzlos entfernen

Alternativ – falls ein eingetragener Seitentitel auf jeden Fall angezeigt werden soll, falls gesetzt: Den o.a. Code ersetzen durch:

```
<?php if ($item['pageTitle']): ?>title="<?= $item['pageTitle'] ?>"<?php
endif ?>
```

Ergebnis:
→ Warnung „Redundanter Titel-Text“ komplett entfernt

Warnung: Text im Blocksatz (Justified text)

Problem: Es gibt eigentlich kein Problem. Der WAVE/AIM-Text weist grundsätzlich auf Blocksatz-Verwendungen hin, weil dadurch z.T. schwer lesbarer Text erzeugt werden könnte – z.B. wenn durch den Blocksatz größere Lücken zwischen Wörtern entstehen.

Falls das bei Ihnen der Fall sein sollte, wenden Sie am besten eine automatische Silbentrennung an, damit längere Wörter harmonischer umgebrochen werden. Nehmen Sie dazu am besten folgende Änderung vor:

Kreuzbund-CSS (files/kreuzbund/css/style.css):

```
neu:
. text-justify p { hyphens: auto; -webkit-hyphens: auto }
```

Warnung: Überschriften-Ebene übersprungen (Skipped heading level)

Problem: Der Text einer Seite sollte in Bezug auf Überschriften-Hierarchien sauber strukturiert sein. Das heißt, die oberste Überschrift sollte eine H1 (Heading 1 = Überschrift 1. Ordnung) sein, danach folgen dann die Überschriften der nächsten Ordnungen (h2, h3, h4, h5 und h6). Zwischen den Überschriften kann natürlich beliebig viel Text und anderer Inhalt stattfinden.

Mitunter werden die Überschriften aber nicht nach Ordnung gesetzt, sondern z.B. nach optischer Gestaltung der Überschrift; in anderen Fällen setzen spezielle Contao-Module Überschriften, ohne dass der Redakteur darauf Einfluss nehmen kann.

Darüber hinaus besteht grundsätzlich der schwierige Fall, Überschriften einer Seite sinnvoll zu strukturieren, die sich in Layout-Elementen befinden, die nicht zum eigentlichen Seitenfluss gehören (z.B. Infoboxen, Seitenspalten, Header- oder Footer-Elemente).

→ Überschriften im Header

Das Logo wird im Basis-Theme leider als Element <h1> ausgegeben. Das ist im entsprechenden Logo-Template so vorgegeben, aber ergibt natürlich semantisch gar keinen Sinn. Daher ändern wir das Template des Logos:

Logo-Template (/templates/oneo/rsce_oneo_logo.html5):

```
alt:

<!-- indexer::stop -->
<h1 class="<?php echo $this->class ?>"<?php echo $this->cssID ?>>
      <a href="<?php echo $this->url ?>">
            <?php if ($image = $this->getImageObject($this->image)): ?>
                  <?php $image->picture['alt'] = $this->alt ?: $image-
>picture['alt'] ?>
                  <?php $this->insert('picture_default', $image->picture)
?>
            <?php else: ?>
                  <?php echo $this->alt ?>
            <?php endif ?>
```

```
      </a>
</h1>
<!-- indexer::continue -->

neu:

<!-- indexer::stop -->
<div class="<?= $this->class ?>"<?= $this->cssID ?>>
      <a href="<?= $this->url ?>">
            <?php if ($image = $this->getImageObject($this->image)): ?>
                  <?php $image->picture['alt'] = $this->alt ?: $image-
>picture['alt'] ?>
                  <?php $this->insert('picture_default', $image->picture)
?>
            <?php else: ?>
                  <?= $this->alt ?>
            <?php endif ?>
      </a>
</div>
<!-- indexer::continue -->
```

Es reicht, die <h1> durch ein Element zu ersetzen; es müssen keine weiteren Anpassungen, z.B. im CSS, vorgenommen werden. Ich habe bei der Gelegenheit noch die „<?php echo“ durch die Kurzschrift „<?=“ ersetzt.

→ Überschriften im Footer

Die Hinweise zu übersprungenen Überschriften-Ebenen beziehen sich zum Teil auf die Footer-Elemente; namentlich die Überschrift „Was können wir für Dich tun“ und die Überschriften im untersten Block „Kreuzbund e.V.“, „Wichtige Seiten“, „Netzwerk“ und „International“.

Lösung:
Um das Problem zu beseitigen, würde ich empfehlen, diese Elemente nicht als Überschrift, sondern als reguläres Element (p, div, span) auszuzeichnen und durch Vergabe entsprechender CSS-Klassen zu gestalten.

Im Falle von kreuzbund.de gebe ich dem Bereich mit den CTA-Buttons die zusätzliche Klasse „footer-buttons“ und ersetze die Texte/Überschriften „Was können wir für Dich tun?“, „Kreuzbund e.V.“, „Wichtige Seiten“, „Netzwerk“ und „International“ durch einfachen fetten Text (strong). Anschließend füge in die CSS-Datei diese Anweisungen ein:

Kreuzbund-CSS (files/kreuzbund/css/style.css):

```
.KBfooter p strong { text-transform: uppercase; font-weight: bold; margin-
top: 2.33em; font-size: 1.1em; }
.footer-buttons strong { font-weight: bold; text-transform: uppercase;
font-size: 1.6em; }
```

→ Überschriften in diversen Template-Elementen

Es gibt einige Inhaltselemente, deren Templates einzelne Felder als Überschrift deklarieren, obwohl es strukturell keinen Sinn ergibt oder im Strukur-Ergebnis zumindest „Glückssache“ ist (vgl. oben, „Timeline“ oder „Icon-Boxen mit Zähler“).

Bei diesen Elementen wurden die Überschriften in den HTML-Template-Dateien durch -Elemente ersetzt und die CSS-Anweisungen entsprechend angepasst.

Beispiel „Icon-Boxen mit Zähler“:

Template (/templates/oneo/rsce_oneo_counter_boxes.html5):

```
alt:
<h3><?php echo $box->name ?></h3>

neu:
<div class="name"><?= $box->name ?></div>
```

Kreuzbund-CSS (files/kreuzbund/css/style.css):

```
neu:
.counter-boxes-item div { font-size: 1.2em; letter-spacing: 0.1em; color:
#656565; text-transform: uppercase; }
.-color-inverted .counter-boxes-item div { color: #fff; }
```

Warnung: Video- und Audio-Elemente (HTML5 video or audio)

Problem: Es gibt eigentlich kein Problem. Der WAVE/AIM-Test weist lediglich darauf hin, dass man diesen Elementen ein Transkript anbeistellen sollte, damit auch entsprechend benachteiligte Personen den Inhalt dieser Medien wahrnehmen können. Der Test überprüft nicht, ob eine solche Datei vorliegt oder korrekt ist – es handelt sich nur um einen allgemeinen Hinweis!

Lösung:
Seit Contao-Version 5.6 gibt es zu diesem Zweck ein eigenes Feld bei den entsprechenden Medien-Elementen („Text-Spur-Dateien“). Dort hinterlegen Sie z.B. eine Datei im VTT-Format („Art der Spur: Closed Captions“). Bitte beachten Sie, das Feld „Titel“ in den Metadaten der .vtt-Datei auszufüllen (Beispiel: „Deutsch“), damit die Transkription im Browser erscheint / ausgewählt werden kann. Sie können bei Bedarf mehrere Transkriptionsdateien je Video hinterlegen.

Für ältere Contao-Version müssten entsprechende Anpassungen in den DCA-, Sprach- und Template-Dateien eingefügt werden.

-- Keine Untertitel im RS Slider?

Leider ist das so. Im Rocksolid-Element „Slider/Video-Slide“ ist das Hinzufügen einer Untertitel-Spur zu einem Video derzeit leider nicht möglich. In den betreffenden Elementen (z.B. „Image-Videos 2026“ auf der Starstseite) muss auf das Ausspielen einer Untertitel-Spur daher vorerst verzichtet werden.

-- Audio-Text wird nicht angezeigt?

Leider ist das so. Mindestens in der Contao-Version 5.7 werden die dem Video-/Audio-Element zugewiesenen Closed-Captions-Spuren nicht im Frontend ausgespielt, falls es sich um ein Audio-Element handelt. Das scheint ein derzeit noch unbestätigter Bug in Contao zu sein, der hoffentlich bald gefixt wird. Falls das Hinzufügen von Closed-Caption-Tracks in Audio-Dateien dringend wichtig ist, müsste dafür ein alternatives Element bemüht werden; z.B. ein HTML-Element oder ein eigenes Custom-Element.

Ich gebe hier mal beispielhaft den Code für eine solche Übergangslösung an:

File „/templates/[ihr-template-verzeichnis)/rsce_kb_audio_config.php“:

```
<?php

return array(
	'label' => array(
		'en' => array('Audio (Temp)', 'Temporary solution for audio files with vtt track'),
		'de' => array('Audio (Temp)', 'Temporäre Lösung für Audio-Files mit vtt-Track'),
	),
	'types' => array('content'),
	'standardFields' => array('cssID'),
	'fields' => array
	(
		'title' => array(
			'label' => array(
				'en' => array('Title', ''),
				'de' => array('Titel', ''),
			),
			'inputType' => 'text',
			'eval' => array(
				'tl_class' => 'w50'
			),
		),
		'audio' => array(
			'label' => array(
				'en' => array('Audio file', ''),
				'de' => array('Audio file', ''),
			),
```

```
                'inputType' => 'fileTree',
                'eval' => array(
                    'fieldType' => 'radio',
                    'filesOnly' => true,
                    'mandatory' => true,
                    //'extensions' => 'mp3,m4a,ogg,wav',
                    'extensions' => 'mp3',
                ),
            ),
            'vtt' => array(
                'label' => array(
                    'en' => array('VTT file', ''),
                    'de' => array('VTT file', ''),
                ),
                'inputType' => 'fileTree',
                'eval' => array(
                    'fieldType' => 'radio',
                    'filesOnly' => true,
                    'mandatory' => true,
                    'extensions' => ' vtt',
                ),
            ),

    ),
);
```

File „/templates/[ihr-template-verzeichnis)/rsce_kb_audio.html5“:

```
<?php

$audioUuid = $this->audio;
$vttUuid = $this->vtt;

$audioUrl = '';
$vttUrl = '';

if ($audioUuid) {
    $audioModel = \Contao\FilesModel::findByUuid($audioUuid);
    if ($audioModel !== null && file_exists($audioModel->path)) {
        $audioUrl = $audioModel->path;
    }
}

if ($vttUuid) {
    $vttModel = \Contao\FilesModel::findByUuid($vttUuid);
    if ($vttModel !== null && file_exists($vttModel->path)) {
        $vttUrl = $vttModel->path;
    }
}
?>
<!-- indexer::stop -->
```

```
<div class="<?= $this->class ?>"<?= $this->cssID ?>>

    <?php if($this->title): ?>
    <p style="font-weight: bold; text-transform: uppercase"><?= $this-
>title ?></p>
    <?php endif ?>

    <figure>
        <audio controls preload="metadata" id="vtt-player-<?= $this->id ?>"
style="width:100%">
            <?php if ($audioUrl): ?>
                <source src="<?= $audioUrl ?>" type="audio/mpeg">
            <?php endif; ?>

            <?php if ($vttUrl): ?>
                <track kind="captions" src="<?= $vttUrl ?>" srclang="de"
label="Deutsch" default>
            <?php endif; ?>

            Dein Browser unterstützt kein HTML5-Audio. Du kannst die Datei
hier herunterladen:
            <?php if ($audioUrl): ?>
                <a href="<?= $audioUrl ?>">MP3 herunterladen</a>
            <?php endif; ?>
        </audio>
    </figure>
    <div style="display:none;text-align:center;font-family:monospace;font-
weight:bold;" id="vtt-text-<?= $this->id ?>"></div>
</div>

<?php if ($vttUrl): ?>
<script>
    (function() {
        var player = document.getElementById('vtt-player-<?= $this->id
?>');
        var textBox = document.getElementById('vtt-text-<?= $this->id ?>');

        if (player && player.textTracks && player.textTracks.length > 0) {
            var track = player.textTracks[0];
            track.mode = "showing";

            player.addEventListener('play', function() {
                textBox.style.display = "block";
            });

            track.addEventListener('cuechange', function() {
                if (this.activeCues && this.activeCues.length > 0) {
                    textBox.innerText = this.activeCues[0].text;
                } else {
                    textBox.innerText = '';
                }
            });
        }
```

```
    })();
</script>
<?php endif; ?>

<!-- indexer::continue -->
```

Verwendung des WAVE/AIM-Tests

Möchten Sie den WAVE/AIM-Test bei Ihrer eigenen Kreuzbund-Website durchführen, so können Sie dazu das Online-Tool unter wave.webaim.org verwenden oder ein entsprechendes Browser-Plugin installieren.

Beachten Sie, dass die Testergebnisse nicht für die gesamte Website gelten, sondern nur für die gerade angezeigte / getestete URL (Seite). Ich erwähne das deshalb, weil ich des öfteren mitbekomme, dass im wesentlichen die Startseite getestet und optimiert wird, und dann davon ausgegangen wird, die Website sei nun entsprechend optimiert. Das gilt dann allerdings nur für die Startseite. Tatsächlich sollte jede einzelne Seite getestet werden, welche noch andere Inhaltselemente enthält als die bis dahin getesteten; im Zweifelsfall eben jede einzelne Seite.

Bitte beachten Sie daher auch, dass sich Bewertungen einzelner Seiten nach deren inhaltlicher/redaktioneller Bearbeitung ändern können, wenn die Seitenredakteur*innen z.B. neue Inhaltselemente eingefügt haben oder bei „altbekannten“ Inhaltselementen formale Fehler gemacht wurden (z.B. vergessen, einen alt-Text für Bilder zu hinterlegen).

Noch ein allgemeiner Hinweis zur „Barrierefreiheit von Websites“

„Barrierefrei“ ist ein Ideal, das man beim Produzieren oder Optimieren einer Website anstreben sollte; wirklich erreichen wird man es eher nie. Zum einen, weil man immer weitere Ansätze finden wird, die Website zu optimieren, denn die Themen sind unerschöpflich: Farben/Kontraste/Helligkeiten, Schriftarten und –größen, Texte und verwendete Sprache (Syntax/Semantik), Sprache, Navigationen, Strukturelle Aspekte (Header, Main, Footer, Hierarchische Gliederung), Mikroformate, Formulare, Alternative Zugangsgeräte, Zeitbasierte Inhalte / Animationen / Effekte, Medien und Dokumente, und vieles mehr. Zum anderen finden ständige Weiterentwicklungen statt, denen gerecht zu werden ist, in Bezug auf Standard, verwendete/weiterentwickelte Softwares/CMS, usw. Es lassen sich verschiedene technische Tests durchführen (wie z.B. der hier verwendete WAVE/AIM-Test), aber viele Aspekte der Barrierefreiheit sind auch eine Frage des Augemaßes oder der individuellen Einschätzung – zum Beispiel in Bezug auf sprachliche Entscheidungen. Daher bleibt die „Barrierefreiheit einer Website“ ein Ideal, an das man sich stets weiter annähern kann.

Ich wünsche Ihnen beim Optimieren Ihrer Website viel Erfolg!

Stand dieses Dokuments: 19.10.2026