Barrierefreie Websites umsetzen mit TYPO3

concept.realisation INTERNETAGENTUR

Claude Unterleitner

- Webprogrammierer (TYPO3 /HTML/ CSS/ PHP) seit 2004, mit Schwerpunkt auf TYPO3
- einer meiner ersten Kunden: Handicap International e.V.: Einrichten von TYPO3 für die damalige Hauptwebsite sowie Projektwebsites, z.B. www.barriere-zonen.org
- Mail: unterleitner@concept-realisation.de

concept.realisation INTERNETAGENTUR

Themen

- Warum barrierefreie Websites?
- Für welche Zielgruppe, technische Grundlagen des W3C
- Tipps für die Erstellung und Strukturierung von Inhalt und HMTL
- TYPO3-Backend-Einstellungen für mehr Barrierefreiheit
- Online-Barrierefreiheit-Tests
- Gefahren einer gut gemeinten, aber falschen Optimierung
- neuere Entwicklungen wie Leichte Sprache

Warum barrierefreie Websites?

„Das Web ist an seiner Basis so gestaltet, dass es für alle Menschen nutzbar ist, unabhängig von ihrer Hardware, Software, Sprache, Kultur, Ort, physischen oder kognitiven Fähigkeiten. Wenn eine Website dieses Ziel erfüllt, dann ist sie zugänglich für ein breitestmögliches Spektrum an Menschen mit unterschiedlichsten Fähigkeiten zu hören, zu sehen, zu verstehen oder sich zu bewegen.“

(Tomas Caspers, www.einfach-fuer-alle.de)

Warum barrierefreie Websites?

- Barrierefreiheit bedeutet, dass Menschen mit Behinderung das Web nutzen können.
- gleichberechtigter Zugang für alle
- lt. der UN-Konvention über die Rechte von Menschen mit Behinderungen ist der ungehinderte Zugang zu Information und Kommunikation **ein grundlegendes Menschenrecht**.
- unterstützt die soziale Inklusion und Integration (überall von jedem nutzbar)

Für welche Zielgruppen?

Antwort: Für Alle!

- Zugang zu Informationen ist einfacher, übersichtlicher, weniger anstrengend
- in bestimmten Situationen hilfreich, z.B. bei langsamer Internetverbindung
- nützlich für Menschen mit Seh- oder Sprachschwierigkeiten, Farbblindheit oder anderen kleineren Schwächen
- nützlich für Suchmaschinen (Google ist blind)

Technische Grundlagen des W3C

- Web Accessibility Initiative des W3C (WAI): Entwicklung von Richtlinien und Techniken, die barrierefreie Lösungen für Web-Software und für Web-Entwickler beschreiben: https://www.w3.org/WAI/standards-guidelines/
- Web Content Accessibility Guidelines und Techniken zeigen detaillierte Informationen für Entwickler. http://www.w3.org/WAI/intro/wcag.php

Tipps für die Erstellung und Strukturierung von Inhalt und HTML

Grundsätzliches zum Inhalt und seiner Struktur:

- Gut lesbare Schrift
- Vernünftige Schriftgröße inkl. Zeilenabstand
- Klare Navigation
- Guter Kontrast
- Eher weniger als mehr grafische Elemente →Fokus klarer für User
- Untertitel für Videos für höreingeschränkte Menschen

Tipps für die Erstellung und Strukturierung von Inhalt und HMTL

HTML:

- Strukturierung der Inhalte h1 … h4, wie bei einer Inhaltsübersicht
- keine leeren (z.B. div oder td) Tags
- Aussagekräftige (!) Alternativtexte (außer Kontext selbsterklärend)
- Link-Title-Tags
- HTML-Reihenfolge sollte einer sinnvollen Anzeige- oder Vorlesereihenfolge entsprechen (z.B. wichtigste Infos zuerst)

Tipps für die Erstellung und Strukturierung von Inhalt und HMTL

HTML:

- Eine Website sollte auch mit ausgeschaltetem Javascript aufrufbar und bedienbar sein
- Eine Website sollte auch nur mit Tastatur benutzbar sein
- Eine Website sollte auch nur mit Maus oder Pad benutzbar sein

Tipps für die Erstellung und Strukturierung von Inhalt und HMTL

HTML:

- barrierefreie Formulare: Formularfelder sollten je Feld immer ein zugehöriges Label enthalten. Wenn ein Label, z.B. für ein Such-Inputfeld, nur für den blinden Nutzer sichtbar sein soll (z.B. wenn Placeholder vorhanden) und sonst nicht, setzt man es am besten per CSS außerhalb des Sichtfelds (z.B. absolut positioniert)
- Tabellen nur dort einsetzen, wo wirklich eine tabellarische Übersicht nötig ist
- Sprunglinks (v.a. um Navigation zu überspringen, wichtig: focus nur wenn aktiv, d.h. erst einblenden/vorlesen, wenn User mit Tabtaste an die Stelle springt)
- Land Mark Roles (z.B.)

ARIA

ARIA = Accessible Rich Internet Applications

- Haben keinerlei Einfluss auf die Darstellung oder das Verhalten einer Webseite
- Sie geben dem Browser oder Screenreader weitere Anwendungsinformationen
- Können als HTML-Element (z.B. <nav> oder als Attribut, z.B. role="nav" verwendet werden.
- Ausführliche Dokumentation unter https://www.w3.org/TR/html-aria/

ARIA

Rolle	HTML-Element
Banner	HEADER-Element
Complementary	ASIDE-Element
contentinfo	FOOTER-Element
Form	FORM-Element mit Bezeichnung
Main	MAIN-Element
Navigation	NAV-Element
region	SECTION-Element mit Bezeichnung
Search	keine HTML-Entsprechung;

ARIA

Bitte beachten:

Wenn ein HTML-Element eine Rolle bereits besitzt, dann darf die Rolle NICHT zusätzlich mit dem role-Attribut zugewiesen werden.

Beispiel für falsch, weil redundant:
<**main** id="content" role="**main**">...</**main**>

ARIA

- aria-label funktionieren wie das title-Attribut, haben aber keinen Tooltip, Beispiel:
  <input aria-label="Suchbegriffe eingeben" name="suche" type="text" />
- aria-labelledby und aria-describedby referenzieren auf das HTML-Element mit der passenden id, z.B.:
  Labeltext blabla

Hinweis: **ARIA wird zwar von einigen Screenreadern unterstützt, aber nicht von allen und auch nicht vollständig. Von den drei Attributen funktioniert derzeit aria-describedby noch am besten.**

ARIA

Die Rollen "form" und "region":

Sowohl bei der Rolle "form" als auch "region" müssen die Seitenregionen mit **aria-label** oder **aria-labelledby** bezeichnet werden, damit sie als Seitenregion identifizierbar sind, z.B. <form id="contact" aria-label="Kontaktformular">

→ **Grund:** nicht jeder Abschnitt und nicht jedes Formular muss oder soll als Region einer Seite ausgezeichnet werden. Ein Kontaktformular könnte bereits identisch mit dem Hauptinhalt (MAIN-Element) der Seite sein.

TYPO3-Backend-Einstellungen für mehr Barrierefreiheit

- Die gute Nachricht: TYPO3 erzeugt von Haus aus bereits weitgehend barrierefreien Code fürs Frontend, keine extra Accessibility Erweiterungen mehr nötig
- Vorsicht bei individuellen Anpassungen!
  Beim Kopieren und Veränderung von (Fluid-) HTML Templates darauf achten, dass der angepasste Code noch barrierefrei ist
  (z.B. Label-Tags auch für neue Formularfelder)!

TYPO3-Backend-Einstellungen für mehr Barrierefreiheit

Redaktionelle Einstellungen:

- Alternativtexte bei Bildern und Link-Title-Tags befüllen (am besten zentral in den Metadaten der Dateien)
- Contentelement Menü: Barrierefreiheitfelder befüllen
- Contentelement Tabelle: Barrierefreiheitfelder befüllen

TYPO3-Backend-Einstellungen für mehr Barrierefreiheit

Barrierefreies Backend?

Ist eines der Ziele der TYPO3 Accessibility Initiative

→ https://bit.ly/2xoSsOp

Online-Barrierefreiheit-Tests

- 128 Tools: https://www.w3.org/WAI/ER/tools/
- Browser-Add-on Wave oder http://wave.webaim.org/
- https://www.einfach-fuer-alle.de/artikel/test-werkzeuge/
- Lesegeräte/Screenreader NVDA:
  Installationsanleitung unter https://bit.ly/2JghxRa
  (leider nur für MS Windows verfügbar)

Zertifizierungsmöglichkeiten

- z.B. durch das Fachzentrum Barrierefreiheit im Internet der Münchner Pfennigparade e.V.:

  https://bit.ly/2xq1VVE

- Beispiel für einen Bewertungsbogen:
  http://testen.bitv-test.de/selbstbewertung/test.php

Gefahren einer gut gemeinten, aber falschen Optimierung

- Extra Textvergrößerungsfunktion, schlimmstenfalls auf der Basis von Javascript:
  ist unnötig, die Browser haben mittlerweile eine sehr gute Textvergrößerungsfunktion, alte Javascriptplugins oft instabil
- Tabindexes, ggf. in Kombination mit Skiplinks, können den Nutzer zu unerwünschten statt der erwarteten Stelle führen (z.B. statt zum weiteren Content zurück zur Navigation)
- Accesskeys haben sich als kontraproduktiv erwiesen

Gefahren einer gut gemeinten, aber falschen Optimierung

- Konflikte zwischen Alttexten und ARIA-Attributen durch die Tatsache, dass ARIA Attribute immer zu einer Fokusierung führen, Beispiel:

```
<li>
    <a aria-label="Diesen Avatar auswählen" href="/#123">
        <img alt="Avatar 123" src="/avatare/123.png">
    </a>
</li>
```

  Wenn ein Screenreader die (gleichartigen) Links nacheinander fokussiert, lauten die Linktexte alle „Diesen Avatar auswählen“ und der Alternativtext wird ignoriert. Diese Ausgabe des Linktextes ist nach HTML-AAM auch korrekt.
  → in so einem Fall nur Alternativtexte verwenden und kein aria-label!

Neuere Entwicklungen wie Leichte Sprache

Leichte Sprache:

- Von zertifizierten Übersetzern erstellen lassen
- Beispiele:
  https://www.leichte-sprache.org/
  Projektwebsite http://www.erschuettert.org
- PDF Beispiel in leichter Sprache:
  https://bit.ly/323gGfk

Quellen und weiterführende Links

https://www.einfach-fuer-alle.de/

https://www.barrierefreies-webdesign.de

https://www.einfach-fuer-alle.de/umsetzen/

https://www.aktion-mensch.de/inklusion/barrierefreiheit/barrierefreie-website.html

http://www.menschzuerst.de/

https://www.bundesfachstelle-barrierefreiheit.de/DE/Themen/EU-Webseitenrichtlinie/eu-webseiten-richtlinie_node.html;jsessionid=7791599B746DCBB6FF8DA86395C57B14

https://www.hellbusch.de/aria-schlaegt-html/

https://www.w3.org/TR/html-aria/

Danke!

concept.realisation INTERNETAGENTUR