



GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Diversity Audit „Vielfalt gestalten“ an der Universität Göttingen

Teilprojekt „Handreichungen zur Barrierefreiheit in der IT“

Abteilung IT

Erstellung barrierefreier Formulare: Beispiele und Erklärungen

Stand Dezember 2019

Die Georg-August-Universität Göttingen hat von 2016 bis 2019 am **Diversity Audit „Vielfalt gestalten“ des Stifterverbandes** teilgenommen. Dieses Dokument ist im Audit-Teilprojekt „Handreichungen zur Barrierefreiheit in der IT“ erarbeitet worden.

Weitere Informationen: www.uni-goettingen.de/diversity-audit

Beispiele und Erklärungen für barrierefreie Formulare

Wahrnehmbarkeit

Unbewusste Änderungen oder Aktionen

Wenn der Fokus nicht über Tab-Taste, sondern Leertaste gesteuert wird, dann kann es zu ungewollten Aktionen kommen. Wenn sich zum Beispiel ein neues Browserfenster öffnet, dann sollte der Fokus nicht ins zweite Fenster wechseln.

Lösung:

- Fokus sollte nur mit Tab-Taste möglich sein
- Fokus darf nicht entfernt werden

Schriftbild

Wird die Schriftvergrößerung genutzt, dann wächst die Schrift nicht standardmäßig mit. Ältere Browser von Opera oder Safari haben mit Angaben wie 100% oder 101% Schwierigkeiten. Die Angabe 1em führt im Internet Explorer bei einer Skalierung zu unverhältnismäßig kleiner oder großer Schrift.

Lösung:

- Relative Schriftgrößenangaben über CSS verwenden
- <input> für einzeiliges Eingabefeld
- <fieldset> für Gruppierung
- <select> für Auswahlliste
- <textarea> für mehrzeiliges Eingabefeld
- <button> für Schaltflächen
- Schriftgröße für Dokumentenkörper den <body>-Bereich in 100.01% festlegen
- Veraltete -Elemente können durch CSS-Attribut font eliminiert werden (so können Angaben zu Schriftart, Schriftgröße und Schriftgewicht eingebunden werden)
- Für skalierbare Schriften besonders geeignet

Beispiel CSS-Datei – Eingabefelder (Ausschnitt):

```
input, select, textarea {  
  font-family: Verdana, Helvetica, Calibri, sans-serif;  
  font-size: 1em;}
```

Beispiel CSS-Datei – Schriftgrößen (Ausschnitt):

```
body.eins { font-size: 100.01%; }
```

Beispiel CSS-Datei – Vererbungsprinzip (Ausschnitt):

```
ul { font: verdana 0.9em normal; }
```

Bedienbar

Postleitzahl falsch und Hausnummer im falschen Feld

Wenn gleich nach der Eingabe der Postleitzahl in eine neue Zeile gewechselt wird, dann kann der Nutzer diese nicht mehr ändern. Beschreibungen für Straße und Hausnummer stehen häufig in einem Label, welches nur dem ersten Input-Element zugeordnet wird. Das Feld für die Hausnummer steht daneben und hat keine Bezeichnung. Daher passiert es, dass neben der Straßenangabe auch fälschlicherweise die Hausnummer eingetragen wird. Diese zu ändern ist dann mit einem erhöhten Aufwand verbunden oder manchmal auch nicht mehr möglich.

Lösung:

- Nach Eingabe der Postleitzahl kein automatisches Springen in ein neues Feld
- Nicht nach Straße, sondern Straßename fragen
- Genaue Zuordnungen der Label und Input-Elemente

Pflichtfelder und Ergänzungen

Es ist nicht immer gleich ersichtlich welche Felder Pflichtfelder sind. Weiterhin werden Angaben ohne Label vom Screenreader nicht erkannt und vorgelesen. Ein Text ohne Label wird dann einfach übersprungen.

Lösung:

- Pflichtfelder mit * oder einem Hinweis wie zum Beispiel erforderlich kennzeichnen
- Nicht relevante Angaben sollten als optional gekennzeichnet werden
- Mit WAI-ARIA-Attribut können erforderliche Felder gekennzeichnet werden
- Mit WAI-ARIA-Attribut können Ergänzungen unterhalb des Eingabefeldes ersichtlich gemacht werden

Beispiel um erforderliche Felder zu kennzeichnen:

```
<input id="vollerName" aria-required="true"/>
```

Vorgehen für Ergänzungen unterhalb des label-Elements:

1. Text mit der weiterführenden Erklärung in ein Element einfassen in dem eine ID über das id-Attribut vergeben wird
2. Dem eigentlichen input-Element das Attribut aria-describedby hinzufügen und als Wert die im ersten Schritt vergebene ID verwenden

Option <optgroup>

Für mehr Barrierefreiheit könnte die Option <optgroup> verwendet werden. Leider wird diese Option nicht von allen Endgeräten oder Screenreadern unterstützt. Wenn <optgroup> nicht unterstützt wird, dann wird es von den Geräten einfach ignoriert. Weiterhin bieten nicht alle Browser eine intuitive Navigation mit der Tastatur für mehrere Auswahlmenüs an. Außerdem haben Personen mit motorischen Beeinträchtigungen teilweise Probleme die richtigen Sachen anzuklicken.

Lösung:

- Option <optgroup> vermeiden
- Mehrere Auswahlmenüs sollten vermieden werden
- Satz von Checkboxen nutzen

Button Attribut

Ein Button Attribut sollte nie leer sein, da der Screenreader dies dann nicht vorliest.

Lösung:

- Keine leeren Button Attribute verwenden
- Bild-Button sollte einen alternativen Text haben
- Absende-Button sollte von der Auswahlliste getrennt sein

Verständlichkeit

Textgestaltung

Bei langen Schachtelsätzen haben sehbehinderte Personen oft Schwierigkeiten dem Satz zu folgen. Dadurch sind sie gezwungen, gegebenenfalls den Screenreader mehrmals für solche Sätze zu verwenden, um den gesamten Inhalt des Textes erfassen zu können. Weiterhin haben Gehörlose mit der schriftlichen Sprache Probleme, da die Primärsprache der meisten gehörlosen Menschen sowie vieler stark Hörgeschädigter die Gebärdensprache ist.

Lösung:

- Text kurz, verständlich sowie übersichtlich gestalten und eine einfache Sprache verwenden
- Klare Anweisungen geben welche Eingaben gefordert werden
- Akronyme sowie Abkürzungen vermeiden
- Text zur Überprüfung selbst laut vorlesen

Abkürzungen und Akronyme

Da es manchmal schwer fällt zu entscheiden, ob es sich um eine Abkürzung oder ein Akronym handelt, sollte immer der Abkürzungs-Tag <abbr> verwendet werden.

Beispiel:

```
<abbr title=„European Community Action Scheme for the Mobility of University Students“>Erasmus</abbr>
```

Sprache

Wenn nicht gekennzeichnet wurde um welche Sprache es sich bei dem Text handelt, dann kann es Probleme mit der Aussprache der Wörter geben.

Lösung:

- Kennzeichnung der Sprache einfügen
- Durch die Nutzung des <lang>-Tag ändert der Screenreader die Sprache und der Text wird richtig vorgelesen

Beispiel:

```
<p>Bitte überprüfen Sie erneut Ihre <span lang= „en“>E-Mail</span>.</p>
```

Fehlervermeidung

Um unnötige Fehler zu vermeiden, sollten Hinweise für eine notwendige Eingabe sowie eine Korrekturhilfe vorhanden sein.

Lösung:

- Fehlerhafte Felder sollten gekennzeichnet werden
- Korrekturen sollten vorgeschlagen werden
- Vor dem endgültigen Absenden sollte eine Übersichtsseite mit Korrekturmöglichkeit vorhanden sein

Robustheit

Nutzung des Formulars

Ein Formular welches nicht komplett angezeigt beziehungsweise vom Screenreader nicht gelesen werden kann, kann auch nicht bearbeitet werden.

Lösung:

- Verwendung von responsive Design
- Eine Kontrolle durch assistive Technologien wie, Screenreader, Steuerung durch Tastatur oder Spracheingabe sollt möglich sein

Eindeutige ID

Screenreader unterstützen die Anzeige von mehreren Labels mit einem Formularelement nicht.

Lösung:

- Nur ein Label zu einem Formularelement
- WAI-ARIA-Attribute verwenden

Weitere Links zum Thema WAI-ARIA finden Sie in den nachfolgend aufgeführten Links:

- <https://www.w3.org/TR/wai-aria-practices/#tabpanel>
- https://developer.mozilla.org/en-US/docs/Learn/Accessibility/WAI-ARIA_basics
- <https://dev.opera.com/articles/ux-accessibility-aria-label/>