Barrierefreie Layouts im Webclient gestalten

- Barrierefreie Layouts
  - How-To: Barrierefreie Layouts im Webclient
    - Toggletipps
      - Vollständige Labels
      - Gliederung der Inhalte
      - Abstände und Responsiveness
      - Definition der Reihenfolge für die Tastaturnavigation
        - Aktive und inaktive Felder
        - Textinhalte mit Hyperlinks
        - Tab-Panels
      - Weitere Hinweise
        - Links in Subforms
        - Verzichten auf Schachtelung scrollender Panels
        - Anleitungen für Nutzer
        - Verzicht auf visuelle Hinweise
        - Alt-Texte
        - Visuell versteckte Elemente

Barrierefreie Layouts

Zukünftig bietet Nuclos alle Voraussetzungen um eine barrierefreie Nutzung des Webclients zu ermöglichen. Barrierefrei heißt in diesem Zusammenhang eine Erfüllung der Kriterien nach BIK-BITV 2.0 (und damit der Norm EN 301 549 sowie der Richtlinie WCAG 2.1). Zum jetzigen Zeitpunkt sind diese Anforderungen nahezu vollständig erfüllt.

Tickets für die erfolgten Anpassungen:

- NUCLOS-8937
- NUCLOS-9815

Offene Punkte:

- Anpassung des Default-Farbschemas für Mindest-Kontrastwerte: NUCLOS-9827
- Vereinheitlichung des Fokushighlightings: NUCLOS-10034

How-To: Barrierefreie Layouts im Webclient

⚠ Manche der hier vorgestellten Techniken werden vom Desktopclient nicht unterstützt oder nicht auf die gleiche Weise dargestellt. Es empfiehlt sich deshalb wenn nötig separate Layouts für den Webclient anzulegen (Eintrag "weblayout" in der Spalte *Benutzerdefinierter Schlüssel* bei der Layoutverwendung) .

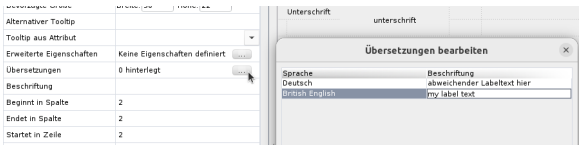
ⓘ Je einfacher die Layouts gehalten werden, desto leichter fällt die Bedienung. Im besten Fall sind die Layouts linear, d.h. eine einfach Abfolge von Feldern, ggf. mit Zwischenüberschriften. Eine barrierefreie Gestaltung von komplexen Layouts ist möglich, aber aufwendiger.

Toggletipps

Anstelle der bekannten Tooltipps mit Popups bei mouseover-Events wird der Inhalt des Tooltippattributs für Layoutelemente jetzt bei Klick auf einen neben dem Feld eingeblendeten Fragezeichenbutton eingeblendet und erst nach einer erneuten Eingabe wieder ausgeblendet.

Vollständige Labels

Für eine barrierefreie Darstellung von Eingabemasken ist es wesentlich, das Labels auf die Eingabefelder verweisen, auf die sie sich beziehen. Dafür wird nun bei Verwendung der Datenkomponenten "mit Label" das *for*-Attribut des erzeugten html-Elements auf die ID des zugehörigen input-Elements gesetzt. Um Labels, deren Text von der Feldbezeichnung des BO abweicht auch in barrierefreien Layouts zu verwenden, muss dieser abweichende Text als Übersetzung für das auf dem oben genannten Weg erzeugte Label hinterlegt werden.



Ohne ein Label können Nutzer, die auf Screenreader, o.a. Hilfstechnologien angewiesen sind womöglich nicht erkennen, welche Funktion ein Feld besitzt.

Elemente vom Typ *Label* sollten in der Regel nie für einen andere Zweck verwendet werden. Für andere statische Textelemente verwenden Sie die neue statische HTML-Komponente.

Gliederung der Inhalte

Um für Hilfstechnologien aufbereitbar zu sein, darf die Gliederung einer Seite nicht nur optisch erfolgen, sondern muss auch programmatisch ermittelbar sein.

Dafür empfiehlt es sich anstelle von *Layoutpanels* mit Titel html-Überschriftenelemente zu verwenden: <h1>, <h2> usw. - so kann eine Gliederung mit bis zu 6 Hierarchieebenen abgebildet werden.

Diese können als statische Komponenten vom typ *HTML Feld* umgesetzt werden. Wenn der default style für das jeweilige h-tag nicht gefällt, kann im Systemparameter custom CSS (siehe Beispiel) eine eigene Klasse mit dem gewünschten Style definiert werden. Der Inhalt des statischen HTML-Feldes könnte dann folgendermaßen aussehen: `<h1 class="myh1class">Dies ist eine Überschrift</h1>`.

Soll auf die bekannten Darstellungselemente nicht verzichtet werden, so kann auch innerhalb eines Titels für ein Layoutpanel html verwendet werden - dies wird jedoch nicht vom Desktopclient ausgewertet.

Abstände und Responsiveness

Die BITV v2 stellt folgende Anforderungen:

> *Die Anpassung der Zeilenhöhe auf das 1,5-fache der Textgröße, des Abstands nach Absätzen auf das 2-fache der Textgröße, von Buchstabenabständen auf das 0,12-fache der Textgröße und von Wortabständen auf das 0,16-fache der Textgröße führt* ***nicht zu einem Verlust an Inhalten oder Funktionalitäten****.*

sowie

> *Inhalte und Funktionen sind bei einer Browserfenster-Breite von 1280px nach Zoom auf 400% (oder bei ungezoomter Ansicht bei einer Browserfenster-Breite von 320px)* ***ohne horizontales Scrollen*** *vollständig erreichbar.*

Beides ist ggf. zu testen und durch die Anpassung von Spaltenbreiten, Zeilenhöhen, Abständen und ggf. der Anordnung der Layoutelemente zu gewährleisten.

Insbesondere ist eine rein vertikale Anordnung von Layoutelementen (statt einer mehrspaltigen Anordnung) zu empfehlen.

Definition der Reihenfolge für die Tastaturnavigation

Für eine fehlerfreie und vollständige Navigierbarkeit des Layouts ohne Zeigegeräte (also mit der Tastur oder per Sprachsteuerung) müssen alle Layoutelemente über die Attribute *Nächstes Feld* oder *Nächste Komponente* miteinander verknüpft werden.

Dabei darf immer nur eines der beiden Attribute verwendet werden und die Feld- und Komponentennamen müssen eindeutig sein. Komponenten können Subforms, Buttons oder statische Inhalte sein.

⚠ Wichtig:

- Der durch diese Verknüpfungen entstehende Graph muss ein geschlossener Kreis sein.
- Alle Laoyutkomponenten müssen im webclient *sichtbar* sein

Einen Sonderfall stellen tab-Panels dar, siehe dafür den entsprechenden Abschnitt.

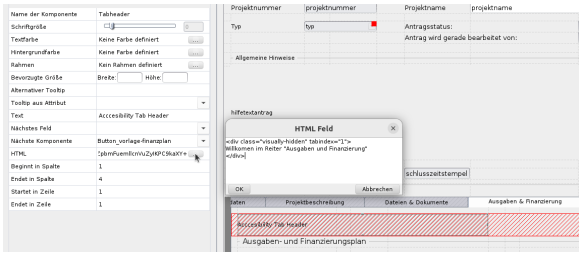
Aktive und inaktive Felder

Auch Felder, die für die Nutzer des Layouts niemals schreibbar sind, sollten in der Regel in die Reihenfolge für die Tastaturnavigation aufgenommen werden. Andernfalls sind Steuerelemente zum Öffnen von Dateien oder navigierbare Links per Tastatur o.a. Hilfstechnologien nicht erreichbar. Außerdem wird so kommuniziert, ob ein Element statisch oder dynamisch ist, was auch dann eine relevante Information sein kann, wenn die Nutzer des Layouts die Inhalte nicht selbst eintragen.

Textinhalte mit Hyperlinks

Ebenso sollten (statische oder dynamische) Texte in die Reihenfolge aufgenommen werden, wenn diese navigierbare Links enthalten sollen. Andernfalls sind diese u.U. nicht erreichbar.

Tab-Panels



Die Verwendung von Tappanels macht das Layout nicht-linear, d.h. der Graph für die Tastaturnavigation bekommt mehrere Zweige. Um mit der derzeitigen Umsetzung trotzdem eine Navigierbarkeit zu gewährleisten, muss das erste navigierbare Element in jedem Tab *denselben* Namen besitzen. Dafür bietet sich die Verwendung eines Dummy-Elements an, das für Nutzer ohne Hilfstechnologien unsichtbar bleibt und zusätzlich weitere Hinweise für Nutzer von Hilfstechnologien enthalten kann. Dies kann mit einer Klasse für visuell versteckte Elemente erreicht werden. Siehe dazu auch den Punkt *Visuell versteckte Elemente* unter dem Abschnitt *Weitere Hinweise.*

Weitere Hinweise

Links in Subforms

Damit Links aus Subforms per Enter geöffnet werden können muss die Spalte den Komponententyp Hyperlink besitzen

Verzichten auf Schachtelung scrollender Panels

Ist für Nutzer, die mit der Tastatuir navigieren und Informationen visuell verarbeiten u.U. nicht bedienbar.

Anleitungen für Nutzer

Um barrierearm zu sein sollten sie Anleitungen bereithalten oder in Ihre Anwendungen integrieren. Darin sollten sie ihr Bedienkonzept erläutern, insbesondere wenn dies von gängigen Patterns abweicht.

Eine Grundlage dafür kann ggf. der Wikiartikel Tastaturnavigation im Webclient bilden.

Verzicht auf visuelle Hinweise

Verzichten Sie in Anleitungen und Hinweisen auf visuelle Indikatoren, ein Negativbeispiel wäre etwa der Hinweis "Klicken Sie auf den grünen Button in der Mitte des Bildschirms.", der gleich zwei visuelle Anhaltspunkte voraussetzt

Alt-Texte

Wenn Bilder oder Icons Inhalte oder Hinweise transportieren sollten diese einen Alt-Text besitzen. Sind die Bilder rein dekorativ sollten sie einen leeren Alttext (alt="") besitzen, der dies signalisiert.

Die Datenkomponente Image besitzt standardmäßig einen leeren Alt-Text. Ist eines der Attribute für Tooltipps ("abweichend" oder "aus Attribut") gesetzt, werden diese auch in den Alt-text übernommen. Es ist also zu beachten, das diese Attribute hier eine Doppelfunktion, besitzen.

Visuell versteckte Elemente

Für visuell versteckte Elemente existiert bereits eine css-Klasse, die im Nuclos-Webclient enthalten ist und für eigene html-Elemente im Layout verwendet werden kann.

```
.visually-hidden{
    position: absolute !important;
    width: 1px !important;
    height: 1px !important;
    padding: 0 !important;
    margin: -1px !important;
    overflow: hidden !important;
    clip: rect(0,0,0,0) !important;
    white-space: nowrap !important;
    border: 0 !important;
}
```

Kontraste

Verwenden Sie von Nuclos-Defaults abweichende Farben oder CSS-Styles sowie visuelle Indikatoren (z.B. Icons oder von Nuclos bereit gestellte Funktionen zur Einfärbung von Zeilen oder Status) beachten Sie auch die Einhaltung der Mindestwerte für den Kontrast:

> *Die für die Identifizierung notwendige visuelle Information von informationstragenden Grafiken und grafischen Bedienelementen sowie deren Zuständen soll einen Kontrast von mindestens 3: 1 zu angrenzenden Farben haben.*

Weitere Informationen hierzu auf der BIK-Homepage .