


Hinweis:

Diese Druckversion der Lerneinheit stellt aufgrund der Beschaffenheit des Mediums eine im Funktionsumfang stark eingeschränkte Variante des Lernmaterials dar. Um alle Funktionen, insbesondere Animationen und Interaktionen, nutzen zu können, benötigen Sie die On- oder Offlineversion. Die Inhalte sind urheberrechtlich geschützt.
©2016 Beuth Hochschule für Technik Berlin

PRT - Prototyp



Lernziele und Überblick

Wenn in der Designphase alle Fragen bezüglich der Farb- und Schriftgestaltung geklärt sind und Sie sich zusammen mit dem Kunden für ein bestimmtes Layout entschieden haben, folgt nun die Erstellung eines Prototypen. „*Prototyping heißt, Designentwürfe und Prototypen immer wieder zu evaluieren und jeden Verbesserungsvorschlag rasch einzubringen und umzusetzen. [...] Solange noch nicht zu viel Aufwand in den Prototypen gesteckt wurde, besteht auch die Chance, schlechte Entwürfe wegzuerwerfen.*“  [MM02]

Bis zu diesem Zeitpunkt lagen alle Ihre Konzepte und Pläne des Seitenlayouts lediglich als statische Grafiken oder Skizzen vor. Sicherlich werden Sie sich schon während des Gestaltungsprozesses viele Gedanken zur späteren Umsetzbarkeit gemacht haben. Möglicherweise haben Sie problematische Teile des Layouts zu Testzwecken in HTML ausprogrammiert, um ihre Realisierbarkeit und ihr Verhalten für den Fall einer Skalierung oder ähnlichem zu testen. Diese Vorgehensweise ist durchaus sinnvoll. Vor allen aufkommenden Problemen wird sie Sie aber in den meisten Fällen leider nicht bewahren können.

Viele Schwierigkeiten treten erst bei speziellen Kombinationen unter bestimmten Gegebenheiten auf. Häufig gibt es auch nur in einzelnen Browsern oder Browserversionen Probleme. Daher stellt auch das Testen Ihrer Webseiten in unterschiedlichen Browsern während der Entwicklungsphase des Prototypen einen wichtigen Arbeitsschritt dar.



Lernziele

In diesem Kapitel lernen Sie:

- wie Sie sich durch die Trennung von Inhalt und Präsentation viel Arbeit und Ärger ersparen.
- wie Sie von nun an Webseiten-Layouts ohne Zuhilfenahme von Tabellen gestalten und welche Vorteile dies mit sich bringt.
- wie Sie nicht-textuelle Objekte barrierefrei in Ihre Webseiten einbinden.
- welche Maßnahmen blinden Internetbenutzern die Navigation im Netz erleichtern.
- welche Probleme von Pop-Ups ausgehen und wie Sie diese lösen können.
- welche Möglichkeiten es gibt, um Ihre Webseiten auf Gültigkeit und Barrierefreiheit zu testen.



Gliederung

Das erste Kapitel dieser Lerneinheit stellt die beiden Arten des Prototypen in der Webentwicklung vor. Anschließend lernen Sie die Handhabung von W3C-konformer Webgestaltung. Darauf aufbauend wird im dritten Kapitel verstärkt auf die Bedeutung von Cascading Stylesheets (CSS) eingegangen.

Wie wichtig ein linearer Aufbau, d. h. eine richtige Reihenfolge beim Erfassen der Webseite durch z. B. Screenreader ist, lernen Sie im darauffolgenden Abschnitt. Die beiden nächsten Kapitel befassen sich mit versteckten und alternativen Texten sowie Textinterlegungen.

Über eine barrierefreie Navigierbarkeit erfahren Sie mehr auf den folgenden Seiten des siebten Kapitels. Barrierefreies Webdesign bezieht sich nicht nur auf die Technik und Form sondern auch auf den Inhalt. Näheres daher zum Umgang mit Abkürzungen, Fremdwörtern und anderen Sprachen können Sie dem Kapitel 8 entnehmen. Auch die Verwendung von Pop-Ups und Links im neuen Fenster (Kapitel 9) und der Einsatz von Formularen (Kapitel 10) sind Themen im Bereich des barrierefreien Webdesigns.

Den Abschluss der Lerneinheit bilden Informationen zu den verschiedenen Tests auf Barrierefreiheit sowie Hinweise zu interessanten Zusatztools. Zum Abschluss der Lerneinheit können Sie wieder Ihr Wissen überprüfen.

1 Art und Umfang des Prototypen

Papier- und funktionale Prototypen

Im Bereich der Webentwicklung gilt es im Wesentlichen zwei Arten von Prototypen zu unterscheiden: Papier-Prototypen und funktionale Prototypen.

Da die Papier-Variante im Hinblick auf die Unterschiede zwischen barrierefreiem und herkömmlichem Webdesign von eher geringer Bedeutung ist, wird auf sie an dieser Stelle nicht weiter eingegangen. Ein Papier-Prototyp sollte in seinem Nutzen dennoch nicht unterschätzt werden. Er eignet sich sehr gut als Vorstufe eines HTML-Prototypen.

Für einen funktionalen HTML-Prototypen ergeben sich unter Berücksichtigung von Barrierefreiheit einige wichtige Aspekte, auf die Sie achten müssen. Dieses Kapitel befasst sich mit der Entwicklung eines funktionsfähigen barrierefreien HTML-Prototypen.




Seitentypen

Damit der Prototyp alle wesentlichen Ansichten und Funktionen der späteren Website verdeutlichen kann, sollte er von jedem Seitentyp zumindest ein Exemplar beinhalten. Zu diesen Seitentypen zählen die Startseite, Strukturseiten (Übersichtsseiten), Inhaltsseiten jeder Navigationsstufe, Formularseiten und so weiter. Hinzu kommen je nach Art des Internet-Angebots weitere Seiten, wie etwa Produkt- und Warenkorb-Darstellungen in einem Online-Shop. Außerdem sollte der Prototyp einen Eindruck von der Druckversion der Seiten vermitteln und eventuell Beispiele für barrierefreie PDF-Dokumente liefern.

2 Trennung von Inhalt und Design


Designvorgaben

Bei der Umsetzung der aufgestellten Designvorgaben in eine Internetseite müssen Sie unbedingt auf eine klare Trennung zwischen Inhalt und Präsentation achten. Seit Erfindung des World Wide Web ist HTML als eine Sprache zur Strukturierung von Inhalten gedacht. In den HTML-Quellcode gehören daher keine Anweisungen bezüglich der Darstellung wie Schriftart, Farbe, Position und so weiter. Für Angaben, die die Formatierung und das Layout betreffen, sieht das W3C in ihren Spezifikationen die Cascading Stylesheets (CSS) vor. Auch die BITV schreibt in Bedingung 3.3 eine Trennung von HTML und CSS vor.  [BITV02]



Vorteile

Diese Trennung bringt eine ganze Reihe von Vorteilen mit sich. Der Seiten Quellcode wird durch sie erheblich übersichtlicher. Alle Formatierungsbeschreibungen lassen sich auf eine externe CSS-Datei auslagern, die von beliebig vielen HTML-Seiten eingebunden werden kann. Änderungen am Design der gesamten Website müssen von Ihnen somit nur an einer zentralen Stelle vorgenommen werden und wirken sich durch die Einbindung auf alle relevanten Seiten aus. Dies erleichtert Ihnen die Wartung erheblich und verkürzt zugleich die Ladezeiten für alle Benutzer.


Externes Stylesheet

Darüber hinaus ermöglicht ein externes Stylesheet erst eine textorientierte Darstellung, welche für die Barrierefreiheit einer Seite erforderlich ist. Bei dieser Darstellung werden alle Stilangaben ignoriert, so dass der Textbrowser oder Screenreader ein sinnvoll ausgezeichnetes Textdokument erhält. Dies funktioniert allerdings nur korrekt, wenn Sie die strukturellen Auszeichnungsmöglichkeiten, die Ihnen HTML bietet, auch richtig einsetzen. Überschriften sollten mit den Tags `<h1>` bis `<h6>`, Listen mit `` oder `` und Zitate mit `<blockquote>` gekennzeichnet werden, um nur drei Beispiele zu nennen. Diese Forderung findet sich auch in den BITV Bedingungen 3.5 bis 3.7 wieder  [BITV02].

3 CSS statt Layouttabellen


In engem Zusammenhang mit der Forderung, keine Angaben zur Formatierung im HTML-Code zu platzieren, steht das Verbot, Tabellen für Layoutzwecke zu verwenden. BITV Bedingung 5.3 besagt: „*Tabellen sind nicht für die Text- und Bildgestaltung zu verwenden, soweit sie nicht auch in linearisierter Form dargestellt werden können.*“  [BITV02] Eine korrekte Linearisierung ist allerdings bei Seiten, die mit Hilfe von Tabellen formatiert wurden, nur selten möglich. „*Layouttabellen sind wie Briefe in Excel*“  [HB05].

Screenreader lesen Tabellen grundsätzlich Zeile für Zeile vor und geben die Zellen von links nach rechts aus. Dadurch können die Inhalte von Zellen, die bei grafischer Darstellung optisch zusammenhängen, im linearisierten Text an ganz unterschiedlichen Positionen stehen. Noch problematischer wird es, wenn für kompliziertere Layouts Tabellen innerhalb von anderen Tabellen eingefügt werden.

Weshalb man heutzutage immer noch sehr viele durch Layouttabellen gestaltete Internetseiten findet, ist unverständlich. „*Historisch gesehen waren Tabellen der ‚Ausweg‘ für fehlende Browser-Kompatibilität bei der Interpretation von CSS.*“  [HB05]. Seit mehreren Jahren jedoch bieten alle gängigen Browser eine weitestgehend korrekte Umsetzung von Cascading Stylesheets. Nach einer kurzen Zeit der Einarbeitung in das Thema CSS lässt sich praktisch jedes Layout ohne Zuhilfenahme von Tabellen realisieren.


Tabellen für
Layoutzwecke

Vorteile der CSS-Formatierung

Neben der Barrierefreiheit bringt dies noch weitere Vorteile mit sich. Verglichen mit einem Tabellen-Layout haben Seiten mit Stylesheets viel kürzere Ladezeiten. Nur wenigen Benutzern ist bewusst, dass Tabellen vom Browser nicht Schritt für Schritt sondern als Ganzes vom Server geladen werden. Daher wird innerhalb einer Tabelle so lange nichts angezeigt, bis ihr Inhalt vollständig geladen ist.  [SY04]

Darüber hinaus vereinfacht der Einsatz von CSS die Wartung einzelner Seiten oder des gesamten Site-Layouts deutlich. Durch die komplexe Menge an HTML-Elementen und die oftmals notwendige mehrfache Verschachtelung kann die Pflege von Tabellenlayouts schnell zum Alptraum werden.

Im Gegensatz dazu gestaltet sich die Wartung von Websites, die mit CSS formatiert sind, viel komfortabler. So können Sie allein durch den Austausch des Stylesheets das Layout einer Seite vollständig verändern, ohne dabei den Inhalt der HTML-Datei zu bearbeiten. Mit der Modifikation weniger Anweisungen in der externen CSS-Datei wird aus einer links ausgerichteten Navigation beispielsweise eine horizontale Navigationsleiste. Dabei wirkt sich diese Änderung einheitlich auf alle Seiten Ihrer Internet-Präsenz aus, die das externe Stylesheet verwenden.

Eine sehr eindrucksvolle Demonstration dessen, was allein durch den Austausch der in einer HTML-Seite verwendeten Stilvorlage möglich ist, bieten die Internetseiten von  [CSS Zen Garden](http://www.csszengarden.com).

3.1 Seitenlayout mit <div>-Containern

Das CSS- Boxmodell

Eine Grundvoraussetzung für die erfolgreiche Umsetzung eines Seitenlayouts durch die Verwendung von CSS ist ein gutes Verständnis des sogenannten Box-Modells. Jedes Element der HTML-Datei wird in einer Box dargestellt. Diese Box ist ohne Stilvorgaben zunächst nicht zu erkennen, da alle Attribute standardmäßig unsichtbar sind. Per CSS lassen sich nun von jedem Element folgende Eigenschaften ändern (siehe Abbildung):

- die Breite und die Höhe des Elementinhalts (`width / height`)
- der Innenabstand (`padding`)
- die Rahmenstärke (`border-width`)
- der Außenabstand (`margin`)

Darüber hinaus können der Inhaltsbereich und der Rahmen durch Farbangaben (`background-color` beziehungsweise `border-color`) sichtbar gemacht werden. Die anderen Bereiche bleiben stets unsichtbar und dienen nur der Formatierung.



Abb.: Das CSS-Boxmodell

Die Darstellung jedes HTML-Elements lässt sich über ein verschachteltes Kasten-System definieren. Der innere Bereich enthält den Inhalt; für ihn können Höhe und Breite festgelegt werden. Umgeben ist der Inhaltsbereich vom Innenabstand, gefolgt vom Rahmen. Den äußeren Bereich bildet der Außenabstand.

Position

Des Weiteren lässt sich die Position jeder Box abhängig oder unabhängig von ihrem Kontext über die Eigenschaften `position` in Kombination mit `top`, `bottom`, `left` und `right` kontrollieren.

Detaillierte Informationen zu den einzelnen Box-Attributen sowie zur Positionierung und Anzeige von Elementen mit Hilfe von CSS finden sie unter anderem bei [www.SELFHTML](#) und in dem Buch „Cascading Stylesheets“ von Shafer und Yank [SY04] .

Container-Layout

Zur Realisierung eines Spalten- beziehungsweise Bereichslayouts mit CSS werden in der Regel die einzelnen Bereiche durch `<div>`-Container angelegt. Diese unsichtbaren „Behälter-Elemente“ nehmen die eigentlichen Seiteninhalte wie Texte, Grafiken, und so weiter auf und können Ihren Anforderungen entsprechend formatiert und positioniert werden. Dazu weisen Sie jeder `<div>`-Box eine eindeutige id zu und stellen im Stylesheet für diese id-Klassen die passenden Breiten, Höhen, Außenabstände und so weiter ein.




Beispiel

Definition eines <div>-Containers HTML-Code

```
...
<div id="navigation">
  <ul>
    <li>Navigationspunkt 1</li>
    <li>Navigationspunkt 2</li>
    <li>Navigationspunkt 3</li>
    <li>Navigationspunkt 4</li>
  </ul>
</div>
...
```

Definition eines <div>-Containers CSS-Code

```
...
#navigation {
  position: absolute;
  top: 100px;
  left: 20px;
  width: 16em;
  height: 24em;
  color: #d7dde4;
  background-color: #1e3d5d;
}
...
```



 [Container-Beispiel anzeigen](#)

Beispiele

Auf www.glish.com/css finden Sie viele Beispiele mit dem dazugehörigen (auf das Nötigste reduzierten) CSS-Code für die gängigsten Spaltenlayouts von Webseiten.

4 Linearisierbarkeit

Uneingeschränkt nutzbar

„Wie auch immer ein Webauftritt mit CSS gestaltet wird, es wird immer Nutzer geben, die Stylesheets gar nicht angezeigt bekommen oder diese ausgeschaltet haben.“  [HB05]. Für diesen Fall ist es wichtig, dass die Website trotzdem uneingeschränkt nutzbar bleibt. So schreibt es auch BITV Bedingung 6.1 vor  [BITV02].


Sinnvolle Reihenfolge

Das bedeutet, dass ein (Text-) Browser oder Screenreader den Inhalt einer Seite in der richtigen, sinnvollen Reihenfolge darstellt. Für einen blinden Benutzer macht es nämlich wenig Sinn, wenn er beim Besuch einer Webseite zuerst die Verweise einer Metanavigation vorgelesen bekommt und erst im Anschluss daran den Seitentitel. Damit ein Besucher sofort weiß, wo er sich befindet, muss er auch als erstes den Titel erfahren.

Üblicherweise beginnen Sie bei der Erstellung des Prototypen damit, das Bereichslayout durch die grobe Einteilung in <div>-Kästen festzulegen und diese erst anschließend weiter zu verschachteln und mit Inhalt zu füllen. Dabei ist es bereits bei der ersten Einteilung wichtig, dass Sie auf eine korrekte Linearisierbarkeit achten. Ein Header-Titel steht in der linearisierten Darstellung nur dann an vorderster Stelle, wenn er auch das erste inhaltliche Element der HTML-Datei ist.

Header-Positionierung

Per CSS lässt sich der Header-Bereich später an jeder beliebigen Stelle positionieren, so dass er in jedem grafischen Browser beispielsweise erst nach einer Navigationsleiste auftaucht. Zur Gewährleistung der Linearisierbarkeit ist jedoch eine korrekte Ordnung des HTML-Codes erforderlich.

Zu einer sinnvollen Reihenfolge der Bereichs-Boxen einer Internetseite empfiehlt Hellbusch folgendes: „Beispielsweise interessieren sich die Besucher einer Seite [...] in erster Linie für die Inhalte. Also sollte sich der Kasten mit dem Inhalt ziemlich am Beginn des Quelltextes befinden. Der Besucher benötigt zudem eine schnell auffindbare Navigation. Inhalte, die sich auf vielen Seiten wiederholen, könnten dahingegen am Ende der Kastenreihe eingebunden werden.“  [HB05].

5 Überspringbare Navigations-Gruppen

Unterteilungen

Nachdem nun also Ihr Seitenlayout als Grundgerüst aus formatierten und positionierten `<div>`-Kästen steht, können Sie dieses durch weitere Unterteilungen verfeinern und mit Inhalt füllen.

Den Benutzern von Screenreadern und Braillezeilen kommt es sehr zu Gute, wenn sie zu Beginn jedes Bereichs eine kurze Beschreibung über dessen Inhalt bekommen. Sehbehinderte Nutzer können sich nicht so leicht wie die Benutzer von grafischen Webbrowsern einen schnellen Überblick über eine Seite verschaffen, um zu erfahren, wo sich die für sie wichtigen Informationen befinden. Stellt man blinden Besuchern keine Hilfen zur Orientierung bereit, müssen sie die gesamte Seite vom Anfang bis zum Ende linear durchgehen, um sie zu erfassen.

Navigation als Liste

Um dies zu verhindern, legt die BITV in Bedingung 12.3 sowie 13.6 fest, dass leicht handhabbare Informationsblöcke geschaffen und inhaltlich zusammenhängende Links gruppiert werden sollen. Es muss möglich sein, diese Gruppen anhand eines Titels zu identifizieren und ganze Navigationsgruppen zu überspringen [BITV02]. Um dieser Forderung nachzukommen, empfiehlt es sich, zusammengehörige Navigationspunkte in einer (unsortierten) Liste zusammenzufassen. Dabei brauchen Sie nicht zu befürchten, dass Ihre Navigationen nun optisch auf das Standard-Listenformat beschränkt sind. Dank CSS ist es möglich, Listen ohne oder mit selbst erstellten Aufzählungspunkten darzustellen und die Einträge vertikal, horizontal oder in beliebiger anderer Form anzuordnen.

Bei [www.w3.org/Listamatic](#) finden Sie eine große Auswahl an interessanten Beispielen dafür, was mit Listen und CSS möglich ist. Zu allen Beispielen wird der Quellcode in übersichtlicher Form mitgeliefert und mehrere Tutorials erklären das schrittweise Vorgehen bei der Erstellung von Navigationslisten.

Sichtbare und unsichtbare Gruppen-Titel

Ob Sie die Bereichs- und Navigationsgruppen-Titel allen Benutzern anzeigen oder nur denen, die keinen grafischen Browser verwenden, bleibt Ihnen überlassen. Häufig werden diese Titel-Elemente mit Hilfe der CSS-Anweisung `display:none`; oder `visibility:hidden`; versteckt, so dass sie nur bei deaktivierten Stylesheets angezeigt werden. Jedoch gibt es Screenreader, die auch CSS interpretieren und den so versteckten Text nicht vorlesen.

Alternative Möglichkeiten

Um dieses Problem zu umgehen, existieren 2 alternative Möglichkeiten:

1. Die Unterbringung der Beschreibung im alt-Attribut einer unsichtbaren Grafik. Dazu legen Sie ein kleines transparentes GIF an und verwenden es mit einer Größeneinstellung von 1x1 Pixel. Der Inhalt des alt-Attributs ist der zu versteckende Bereichstitel. Dieser wird nur bei ausgeschalteten Grafiken, wie es bei Screenreadern und Textbrowsern der Fall ist, angezeigt.



Beispiel

```

```

2. Ein Verstecken der Beschreibung durch eine Positionierung außerhalb des Browserfensters. Um dies zu erreichen, weisen Sie dem Element, das die Beschreibung enthält, per CSS eine absolute Position außerhalb des sichtbaren Bereichs zu. Am besten geht dies durch die Angabe von negativen Werten für die Eigenschaften `top` oder `left`.



Beispiel

```
<p style="position:absolute; top:-200px;">Es folgt die Metanavigation</p>
```

Überspringbarkeit

Selbstverständlich funktioniert dies auch mit einem externen Stylesheet. Nur bei deaktiviertem CSS wird das Element in der linearisierten Darstellung an seinem eigentlichen Platz im Quelltext angezeigt. Da für einen Screenreader die, durch das Stylesheet bestimmte, Position keine Rolle spielt, wird es von ihm auch an der entsprechenden Stelle im Quellcode vorgelesen.

Skip-Links

Die ebenfalls von der BITV geforderte Überspringbarkeit lässt sich über einen „Skip-Link“ realisieren. Ein Link zu Beginn der Seite, den Sie ebenfalls - wie oben beschrieben - vor grafischen Browsern verstecken können, verweist auf den Startpunkt des Inhalts oder auf den Anfang einer bestimmten Navigationsgruppe.




Beispiel

```
<a href="#content" title="Überspringt die Navigationsbereiche"
class="invisible">Direkt zum Inhalt springen</a>
...
<div id="content">
    Hier steht der Inhalt
    ...
</div>
```

6 Texthinterlegung

Informationen
uneingeschränkt erfassbar

Bei der Einbindung des Inhalts in Ihre Internetseiten muss gewährleistet sein, dass alle Informationen, die als Grafiken oder in anderer multimedialer Form vorliegen, uneingeschränkt erfassbar sind. Wie bereits in Lerneinheit „ANA - Anforderungsanalyse“ und Lerneinheit „LAD - Layout und Design“ erwähnt wurde, müssen Sie in Einklang mit BITV Anforderung 1 jedes Element, welches kein Text ist, mit einer inhaltlich gleichwertigen textuellen Beschreibung hinterlegen  [BITV02]. Die am häufigsten auftauchenden Elemente, die einen Alternativtext erfordern, sind sicherlich Grafiken. Aber auch Java-Applets, Flash-Objekte, Filme und Audio-Objekte müssen mit Beschreibungen hinterlegt werden. Dabei reicht es nicht aus, beispielsweise einen Video-Clip mit dem `alt`-Text „Das Video zeigt den natürlichen Wasserkreislauf in einer Animation“ zu versehen. Wenn dieser Wasserkreislauf im Video ausführlich erläutert wird, muss auch der beschreibende Text diese Informationen vermitteln.

Insbesondere bei Grafiken, die der Navigation dienen, muss darauf geachtet werden, dass in der Regel nicht der Inhalt sondern ihre Funktion entscheidend ist. So wäre etwa der `alt`-Text eines „Home-Buttons“ mit „Die Grafik zeigt ein Haus-Symbol“ schlecht gewählt. Ein korrekter Alternativtext müsste „zur Startseite gehen“ lauten.

Für kleine Grafiken, die keine komplizierten Sachverhalte oder Abläufe beschreiben, reichen die empfohlenen 150 Zeichen des `alt`-Attributs meist aus. Muss hingegen der Inhalt eines komplexen Ablaufdiagramms oder die Textabschrift einer Video- oder Sprachdatei zur Verfügung gestellt werden, sollten Sie auf eine der folgenden Methoden zurückgreifen.

6.1 longdesc-Attribut und D-Links

Alt-Attribut

Für den Fall, dass ein Objekt eine ausführliche Erläuterung als Alternativtext benötigt, sollten Sie im `alt`-Attribut nur eine kurze Zusammenfassung angeben und die detaillierte Beschreibung mit Hilfe des `longdesc`-Attributs oder durch einen D-Link bereitstellen.

longdesc-Attribut

`longdesc` lässt sich - wie auch das `alt`-Attribut - in jedem `img`-Tag anwenden. Nur enthält es als Wert nicht den eigentlichen Beschreibungstext, sondern einen Verweis auf eine (X)HTML- oder TXT-Datei mit dem Text. Empfehlenswert ist es, dafür entweder eine reine Textdatei oder eine (X)HTML-Seite ohne besonderes Layout zu verwenden.

Die meisten Screenreader können `longdesc`-Beschreibungen darstellen. Die Benutzer grafischer Webbrowser hingegen bekommen davon normalerweise nichts mit, es sei denn, sie lassen sich in einem Mozilla-basierten Browser die Eigenschaften eines Bildes anzeigen.

D-Links

Eine andere Möglichkeit zur Bereitstellung langer Beschreibungen ist die Verwendung eines sogenannten D-Links. Das „D“ steht für Description, also zu deutsch Beschreibung. Unmittelbar neben der Grafik oder dem zu erklärenden Objekt fügen Sie einen Link, der lediglich aus dem Text „D“ besteht, ein. Dieser Link sollte zu einer gesonderten Seite führen, auf der die ausführliche Erläuterung zu finden ist. [1]

Welche der beiden Varianten Sie nun benutzen, bleibt Ihnen überlassen. Aus gestalterischer Sicht kann sich die Unterbringung eines D-Links manchmal unvorteilhaft auf das Seitenlayout auswirken. Es ist aber auf jeden Fall die sicherste Vorgehensweise, da sie mit jedem Browser und Screenreader funktioniert. Ein Argument, das gegen die Verwendung von `longdesc` sprechen könnte, ist die Tatsache, dass dieses Attribut noch nicht von allen Browsern (zum Beispiel dem Internet Explorer) unterstützt wird.

[1]  <http://aktuell.de.selfhtml.org> (Stand: 26.6.2006)

7 Navigierbarkeit


Verständliche Navigierbarkeit

Um die uneingeschränkte Bedienbarkeit einer Webpräsenz zu garantieren, ist eine schlüssige, schnell verständliche Navigierbarkeit unerlässlich. Neben einer sinnvollen Anordnung der Navigationselemente und -bereiche, auf die bereits in Lerneinheit „ANA - Anforderungsanalyse“, Kapitel 5 Sitearchitektur und Navigationsstruktur eingegangen wurde, ist auch die korrekte Benennung von Links wichtig. Außerdem müssen Sie darauf achten, keine redundanten Links auf die gerade angezeigte Seite anzubieten. Sie verwirren die Benutzer nur, da ein Anklicken keine Veränderung bewirkt.



7.1 Eindeutige Linkbezeichnungen

Ziel eines Links eindeutig identifizierbar

In Bedingung 13.1 wird gefordert, dass jedes Ziel eines Links eindeutig identifizierbar sein muss  [BITV02]. Diese Möglichkeit der Identifikation muss auch dann noch gegeben sein, wenn der Kontext eines Links nicht bekannt ist. Benutzer von Screenreadern, die mit der Tabulator-Taste von Link zu Link springen, bekommen für jeden Verweis lediglich den Link-Text und eventuell den title-Wert vorgelesen. Wenn der Nutzer an dieser Stelle nur die Information „mehr“ oder „Klicken Sie hier“ erhält, hilft ihm dies nicht weiter. Daher sollten Sie grundsätzlich aussagekräftige Bezeichnungen als verlinkten Text verwenden und diese - falls erforderlich - durch das title-Attribut näher erläutern.





Beispiel

```
<a href="navibarrierefrei.htm" title="hier erhalten  
Sie nähere Informationen zum Thema barrierefreie  
Navigation">barrierefreie Navigation</a>
```

7.2 Tabindex

Erreichbarkeit aller Links

Neben der Verständlichkeit muss auch die Erreichbarkeit aller Links einer Seite gegeben sein, und zwar unabhängig vom Eingabegerät (BITV Anforderung 9)  [BITV02]. „Auch wenn es weitere Eingabegeräte gibt, so ist die geräteunabhängige Eingabe durch die Sicherstellung der Tastaturbedienbarkeit gewährleistet.“  [HB05]. Ein Benutzer, der seinen Webbrowser ohne Maus, sondern nur über die Tastatur bedient, wird ein Untermenü, das sich erst bei dem JavaScript-Event onmouseover öffnet, niemals erreichen können.


Die Möglichkeit, alle per HTML ausgezeichneten Links einer Webseite der Reihe nach mit Hilfe der Tabulator-Taste anzuspringen, ist in jedem Browser standardmäßig gegeben. Sollte diese vorgegebene Tabulatorreihenfolge nicht sinnvoll sein, lässt sie sich über das Link-Attribut `tabindex` beliebig anpassen:



Beispiel

```
<ul>
  <li><a href="#" tabindex="4">Link1</a></li>
  <li><a href="#" tabindex="1">Link2</a></li>
  <li><a href="#" tabindex="2">Link3</a></li>
  <li><a href="#" tabindex="3">Link4</a></li>
</ul>
```

Mögliche Probleme durch `tabindex`


Für das Überspringen von Navigationsbereichen, um so direkt zum Inhalt zu gelangen, eignet sich `tabindex` jedoch nur bedingt. Da mit der Tabulator-Taste ausschließlich Links und Formularelemente angesprochen werden können, müsste der Inhaltsbereich mit einem dieser beiden Elemente beginnen. Um diese Funktionalität zu erreichen, sind die im Abschnitt  Überspringbare Navigations-Gruppen vorgestellten Skip-Links besser geeignet. Verwendet man sie allerdings in Kombination mit der Hinterlegung von Links durch `tabindex`, kann dies die Gebrauchstauglichkeit beeinträchtigen: [2]

Für den Fall, dass Sie am Anfang einer Internetseite einen „zum Inhalt springen“-Link anbieten, sollte dieser auch den `tabindex`-Wert 1 erhalten. Möchten Sie nun den Links der darauf folgenden Hauptnavigation die `tabindex`-Werte 2, 3, 4 und 5 zuordnen, kommt es zu Problemen mit der Tab-Reihenfolge:


Ein Benutzer, der per Tabulator-Taste den Skip-Link (`tabindex="1"`) auswählt und so zum Inhalt gelangt, möchte nun den ersten Link des Textes im Inhalt auswählen. Durch erneutes Drücken der Tab-Taste gelangt er allerdings wieder an den Anfang der Seite - nämlich zum ersten Link der Hauptnavigation (`tabindex="2"`). Der Skip-Link wird dadurch praktisch unbrauchbar. Dieses beschriebene Verhalten ist sowohl bei Mozilla-, als auch bei Netscape- und Opera-Browsern zu beobachten.

Die beste Möglichkeit ist, dieser Problematik aus dem Weg zu gehen, indem Sie die Navigationspunkte so anlegen, dass bereits das Standardverhalten der Tabulator-Taste eine sinnvolle Reihenfolge ergibt. Die Verwendung von `tabindex` sollten Sie auf die Fälle beschränken, wo es unbedingt erforderlich ist. „Sinnvoll erscheint der Einsatz des `tabindex`-Attributes in Formularen, wenn die Reihenfolge, in der Kontrollelemente bedient oder Texteingaben gemacht werden sollten, nicht der Reihenfolge dieser Elemente im Quelltext entspricht.“ [3]

[2]  www.einfach-fuer-alle.de (Stand: 26.6.2006)

[3]  www.einfach-fuer-alle.de (Stand: 26.6.2006)

7.3 Access-Keys

Eine weitere Maßnahme zur Unterstützung der Tastatursteuerung wird in BITV Bedingung 9.5 gefordert. „*Es sind Tastaturkurzbefehle für Hyperlinks, die für das Verständnis des Angebots von entscheidender Bedeutung sind [...], Formularkontrollelemente und Gruppen von Formularkontrollelementen bereitzustellen.*“  [BITV02]

Über die sogenannten Access-Keys lassen sich den gleichen Elementen, die auch tabindex unterstützen, Tastaturkürzel zuordnen. Ein Betätigen dieser Tasten bewirkt den direkten Sprung zur verknüpften Adresse beziehungsweise zum entsprechenden Formularelement. Zu den beschriebenen verständnisfördernden Seiten gehören etwa Hilfeseiten, das Glossar und die Sitemap.

Das Attribut accesskey erhält als Wert einen Buchstaben oder eine Zahl, die - je nach Browser - zusammen mit einer oder mehreren Steuerungstasten gedrückt werden muss: [4]


Aktivierung von Access-Keys in verschiedenen Browsern:


Browser	Tastenkombination
Internet Explorer	Alt + Access-Key + Enter
Mozilla	Alt + Access-Key
Opera	Shift + Esc + Access-Key
Internet Explorer Mac	Ctrl + Access-Key + Enter
Mozilla Mac	Ctrl + Access-Key

Tab.: Aktivierung von Access-Keys

Mögliche Probleme von Access-Keys

Ein Problem stellt aber die Tatsache dar, dass viele der möglichen Tastenkombinationen in den unterschiedlichen Browsern schon vergeben sind und somit als Schnittmenge der verfügbaren Zeichen eigentlich nur k, r, u, 3 und 8 übrig bleiben. Außerdem ist es jedem Webentwickler selbst überlassen, wie er welche Tasten belegt. Ein Benutzer muss daher bei jeder Seite, die er besucht, die Belegung der Access-Keys neu erlernen. Zudem stellt das gleichzeitige Drücken mehrerer Tasten für Menschen mit motorischen Behinderungen - für die Access-Keys ursprünglich gedacht waren - oft eine unüberwindbare Hürde dar. [5]

[4]  www.einfach-fuer-alle.de (Stand: 26.6.2006)

[5]  www.einfach-fuer-alle.de (Stand: 26.6.2006)

7.4 Sitemap

Inhaltsverzeichnis

BITV Bedingung 13.3 fordert für jeden Internet-Auftritt, eine Übersicht der allgemeinen Anordnung in Form eines Inhaltsverzeichnisses beziehungsweise einer Sitemap bereitzustellen [BITV02]. So eine Gesamtübersicht hilft dem Benutzer bei der Orientierung innerhalb des Auftritts. Dies kann besonders bei großen, unübersichtlichen Sites, die aus vielen strukturell unterschiedlichen Bereichen bestehen, erforderlich sein.

Üblicherweise stellt die Gesamtübersicht die hierarchische Anordnung aller Seiten der Website als Baumstruktur dar (siehe Abbildung). Jeder Eintrag ist als Link anzulegen, der direkt auf die entsprechende Unterseite verweist.



Abb.: Sitemap der Restaurant-Website

Die Abbildung zeigt einen Ausschnitt der Gesamtübersicht. Die Text-Links zu den jeweiligen Unterseiten sind entsprechend der Sitestruktur hierarchisch in einem Baumdiagramm angeordnet.

Suchmöglichkeit

Falls die Gesamtübersicht nicht das bietet, was der Besucher sucht oder aber zu umfangreich ist, um einen schnellen Überblick zu geben, empfiehlt Hellbusch die Bereitstellung einer zusätzlichen Suchmöglichkeit. Dies kann in Einklang mit Bedingung 13.7 entweder eine Filtermöglichkeit für die Sitemap oder eine Volltextsuche sein. Ebenso wäre ein Stichwortverzeichnis oder Glossar eine Möglichkeit für eine strukturunabhängige Übersicht.

8 Verständlichkeit

Einfache Sprache

Bereits in Lerneinheit „ANA - Anforderungsanalyse“ wurde auf die Verwendung einer möglichst klaren, einfachen Sprache hingewiesen. Um die Verständlichkeit von Texten im Internet darüber hinaus weiter zu fördern, verfügt HTML über einige Sprachelemente, die bei der Verdeutlichung und Erklärung sprachlicher Besonderheiten hilfreich sind. Der Einsatz dieser Elemente wird von der BITV in Anforderung 4 verlangt [BITV02].



8.1 Abkürzungen

Abkürzungen und Akronyme erläutern

Laut Bedingung 4.2 müssen Sie Abkürzungen und Akronyme bei ihrem ersten Auftreten im Inhalt erläutern und kenntlich machen [BITV02]. (X)HTML sieht dazu die Elemente `abbr` und `acronym` vor. Mit ihnen lassen sich die entsprechenden Begriffe kennzeichnen (Standard ist in den meisten Browsern eine Unterstrichelung) und über das `title`-Attribut erklären.

abbr und acronym

Über den Sinn und Unsinn der Verwendung von `abbr` und `acronym` wird vielerorts heftig diskutiert, da es eine Menge Probleme und Unstimmigkeiten im Zusammenhang mit diesen Elementen gibt. Was jedoch feststeht, ist folgendes:

- Auch wenn ein semantischer Unterschied zwischen einer Abkürzung (abbreviation) und einem Akronym besteht, bieten beide Elemente dieselbe Funktionalität in der Zugangssoftware. [6]
- Der Internet Explorer unterstützt `abbr` nicht. Wenn also allen Nutzern die Erläuterungen von Abkürzungen zur Verfügung stehen sollen, darf dafür nur `acronym` verwendet werden. [7]
- Im allgemeinen Sprachgebrauch verbreitete Abkürzungen wie Nr., z. B. oder usw. sind zum einen für jeden verständlich und sollten zum anderen im Web sowieso ausgeschrieben werden, da sie eigentlich nur dort benötigt werden, wo wenig Platz zur Verfügung steht. [8]
- Abkürzungen, die in den gängigen Sprachgebrauch eingegangen sind und mittlerweile selbst für die Sache stehen, müssen nicht erläutert oder ausgeschrieben werden. Bei Begriffen wie DVD, NATO oder Radar ist jedem klar, worum es geht. Die eigentlich dahinter stehende Ausschreibung ist hingegen häufig gar nicht bekannt. [9]
- Die ausgeschriebene Form einer Abkürzung aus einem Fachgebiet, das dem Leser nicht geläufig ist, bringt ihm auch keinen Nutzen. Wer nicht weiß, was „CSS“ ist, wird auch mit dem Begriff „Cascading Stylesheets“ nichts anfangen können. Hier kann nur ein Glossar weiterhelfen. Für jemanden aus dem jeweiligen Fachgebiet tritt hingegen der vorherige Fall ein. [10]
- Der einzige Fall, bei dem die Erläuterung von Abkürzungen Sinn hat, ist, wenn die Abkürzung dem Leser unbekannt ist, die Ausschreibung jedoch aussagekräftig ist. Vielen Internetnutzern sagt die Abkürzung „BITV“ nichts; mit den einzelnen Begriffen „barrierefrei“, „Informationstechnik“ und „Verordnung“ können sie jedoch etwas anfangen. [11]

Jedoch stellen nicht alle Screenreader und Textbrowser das `title`-Attribut dar und auch Benutzer grafischer Browser, die keine Maus benutzen, bekommen den Wert von `title` nicht als Tooltip angezeigt. Daher empfiehlt Colin Lieberman in seinem Artikel bei „A List Apart“, Abkürzungen zusätzlich mit einem Verweis auf die Erklärung in einem Glossar auszustatten. [12]

- [6]  www.barrierefreies-webdesign.de (Stand: 15.6.2006)
- [7]  www.barrierefreies-webdesign.de (Stand: 15.6.2006)
- [8]  www.bik-online.info (Stand: 15.6.2006)
- [9]  www.bik-online.info (Stand: 15.6.2006)
- [10]  www.bik-online.info (Stand: 15.6.2006)
- [11]  www.bik-online.info (Stand: 15.6.2006)
- [12]  www.alistapart.com (Stand: 15.6.2006)

8.2 Fremdwörter

CSS-Klasse

Häufig findet man im Web Seiten, auf denen die HTML-Elemente `abbr` oder `acronym` für die Übersetzung oder Erklärung von Fremdwörtern und Fachbegriffen zweckentfremdet werden. Dies ist jedoch nicht nötig. Sowohl die Kennzeichnung, dass es sich um einen Begriff handelt, für den eine nähere Beschreibung angeboten wird (etwa durch Unterstrichelung), als auch die Bereitstellung dieser Beschreibung im `title`-Attribut lassen sich auch über das ``-Element und eine selbst angelegte CSS-Klasse erreichen.



Beispiel

Erläuterung von Fremdwörtern


`<!-- HTML-Code: -->`

```
<p>Das title-Attribut wird von dem <span class="fachbegriff"
title="Programm, welches Internetseiten vorliest">Screenreader</span>
auf Wunsch wiedergegeben.
</p>
```

```
/* CSS-Code: */

.fachbegriff {
    cursor:help;
    padding-bottom:1px;
    border-bottom:1px dotted #000000;
}
```


Probleme

Ähnlich wie bei den  Abkürzungen kann es auch hier zu Problemen bei der Darstellung des `title`-Attributs kommen. Deshalb bietet es sich besonders für Fremdwörter an, diese mit einem Glossar zu verlinken, in dem sie ausführlich erläutert werden.

8.3 Sprachwechsel

Sprachwechsel kennzeichnen

Die meisten Internet-Auftritte verwenden durchgehend dieselbe Sprache. Weichen Sie jedoch im Text von dieser Hauptsprache ab, müssen Sie dies kenntlich machen. Solch ein Sprachwechsel kann etwa beim Zitieren eines fremdsprachigen Textes oder bei der Verwendung von Fachbegriffen auftreten.

„*Sprachausgaben arbeiten mit Sprachtabellen, das heißt, sie erfassen einen Text und geben ihn entsprechend der voreingestellten Aussprache an die Sprachsynthese weiter.*“  [HB05]. Das Programm zur Sprachausgabe kann jedoch bei einem Text nicht selbst erkennen, in welcher Sprache er verfasst ist und liest deshalb nicht gekennzeichnete Fremdwörter falsch vor.




Audioclip

Falsche und korrekte Wiedergabe bei Sprachwechsel

Der folgende Text wurde zweimal durch den Screenreader JAWS vorgelesen; zunächst ohne besondere Kennzeichnung der Sprache, danach mit ausgezeichneten Sprachwechseln im Quelltext:

„*In seinem Artikel bei A List Apart schreibt Colin Lieberman: „AAA-level compliant pages meet the needs of every group of users.“*“

(Quelle des Zitats:  www.alistapart.com/articles/hatrick)


Audioclip: Keine Kennzeichnung der Sprachwechsel

Die meisten englischen Wörter des Textes werden von JAWS falsch ausgesprochen.

Audioclip: Mit Kennzeichnung der Sprachwechsel

Der Text wird von JAWS korrekt vorgelesen. Bei jedem Wechsel der Sprache ist jedoch eine kurze Sprechpause hörbar.

Kennzeichnung der Hauptsprache

Die BITV schreibt sowohl die Kennzeichnung der Hauptsprache (Bedingung 4.2) wie auch die von Sprachwechseln (Bedingung 4.1) vor  [BITV02]. Dazu existiert das HTML-Attribut lang, welches in jedem HTML-Tag verwendet werden kann.



Beispiel

Kennzeichnung von Hauptsprache und Sprachwechseln

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
  <head>
    ...
  </head>
  <body>
    ...
    <p>Die Speisekarte des <span lang="es">El Pabellón</span>
      steht Ihnen als PDF-Datei zum <span lang="en">Download</span>
      bereit.</p>
    ...
  </body>
</html>
```

Gebräuchlich ist es, die Hauptsprache im HTML-Element anzugeben und für die Kennzeichnung einzelner Ausdrücke im Inhalt das ``-Tag zu verwenden. Bei ganzen Absätzen in einer anderen Sprache kann diese für das zugehörige `<p>`-Element angegeben werden.

9 Pop-Ups / Links im neuen Fenster

Pro und Contra

Sucht man im Web nach den Begriffen „barrierefrei“ und „Pop-Ups“, so stößt man auf eine Vielzahl von Seiten, auf denen Verfechter und Gegner von Pop-Up-Fenstern ihre Argumente darlegen. An dieser Stelle werden Ihnen einmal die Argumente aufgelistet, die für und gegen das Öffnen eines Links in einem neuen Fenster sprechen.


Contra Pop-Ups

Gegen Pop-Ups spricht:

- Ältere Screenreader können sich neu öffnende Browser-Fenster teilweise nicht erkennen. [13]
- Im Browser installierte Pop-Up-Blocker können das Erscheinen von Fenstern unterbinden, die nur durch JavaScript-Befehle geöffnet werden.
- Ungeübte Benutzer sind es gewohnt, jederzeit über den „Zurück“-Button auf die vorherige Seite zu gelangen und werden möglicherweise durch neue Fenster verwirrt.
- Viele Internetnutzer möchten selbst darüber entscheiden, ob sich ein neues Fenster öffnet oder nicht und wissen, wie sie dieses Verhalten in ihrem Browser erreichen können. [14]

Pro Pop-Ups

Für Pop-Ups spricht:

- Die BITV rät zwar dazu, Pop-Ups zu vermeiden, verbietet ihren Einsatz aber nicht explizit, sofern der Nutzer darüber informiert wird (Bedingung 10.1).  [BITV02]
- Neuere Screenreader und Vergrößerungssysteme sind in der Lage, den Fokus auf ein sich öffnendes Fenster zu lenken und schwächen die BITV-Forderung somit ab. [15]
- Es gibt tatsächlich Szenarien, die sich mit Pop-Ups besser lösen lassen, wie beispielsweise eine Online-Hilfe Funktion. [16]

Fazit

Fazit: Pop-Up, wenn es sinnvoll ist

„Wie (fast) alles im Web kann diese Technik sinnvoll angewendet werden, reine Spielerei sein oder aber schlichtweg ‚missbraucht‘ werden.“ [17]

„Automatische Pop-Ups, die sich von alleine öffnen, sind weder barrierefrei noch benutzerfreundlich und sollten vermieden werden.“ [18]

Sollte es wirklich nötig und sinnvoll sein, einen Link in einem neuen Fenster zu öffnen, dann informieren Sie den Nutzer im Text, der zu dem Link gehört, darüber. Das title-Attribut reicht dazu allein nicht aus. [19]

Verlassen Sie sich nicht allein auf JavaScript, um ein neues Fenster zu erzeugen.

Verwenden Sie zusätzlich unbedingt das href-Attribut, um das verlinkte Ziel zu definieren und setzen Sie target auf den Wert „`_blank`“. So können Sie sicher sein, dass der Link auch bei deaktiviertem JavaScript funktioniert. Damit Sie das Attribut `target` verwenden können, muss die Dokumententyp-Definition auf „transitional“ eingestellt sein - „strict“ unterstützt dies nämlich nicht.

So wie im folgendem Beispiel könnte der Code für ein Pop-Up Fenster aussehen, welches allen Anforderungen an die Barrierefreiheit entspricht. Unterstützt der Browser eines Benutzers JavaScript, öffnet sich die Hilfeseite in einem Pop-Up mit festgelegter Größe und ausgeschalteten Zusatzleisten. Ist JavaScript im Browser eines Besuchers nicht verfügbar, öffnet sich ein neues Browserfenster beziehungsweise ein neuer Tab (je nach Browser-Einstellung).




Beispiel

Barrierefrei gestaltetes Pop-Up Fenster

```
<html>
  <head>
    ...
    <script type="text/javascript">
      var popupFeatures = "toolbar=no, status=no, menubar=no,
        scrollbars=yes, resizable=yes, width=560, height=400";
      function popup(file) {
        popupWindow = window.open(file, "popWin", popupFeatures);
        popupWindow.focus();
        return false;
      }
    </script>
  </head>

  <body>
    ...
    <a href="hilfe.html" target="_blank"
      onclick="return popup(this.href);" title="öffnet die
      Hilfeseite in einem neuen Browserfenster">zur Hilfeseite
      (öffnet ein neues Fenster)</a>
    ...
  </body>
</html>
```

 Pop-Up Script testen (öffnet ein neues Fenster)

-
- [13]  www.barrierefreies-webdesign.de (Stand: 12.6.2006)
 - [14]  www.webdesign-in.de (Stand: 12.6.2006)
 - [15]  www.barrierefreies-webdesign.de (Stand: 12.6.2006)
 - [16]  www.einfach-fuer-alle.de (Stand: 12.6.2006)
 - [17]  http://barrierefrei.e-workers.de/know_popups.php (Stand: 13.6.2006)
 - [18]  www.bik-online.info (Stand: 12.6.2006)
 - [19]  www.webdesign-in.de (Stand: 12.6.2006)

10 Formulare

Anforderung

„Da über die meisten Formulare eine Interaktion zwischen dem Benutzer und dem Anbieter stattfindet (Kauf, Kontaktaufnahme, Recherche, und so weiter) sollte es eigentlich im ureigensten Interesse des Anbieters liegen, allen möglichen Ausgabeformaten und Formen der Bedienung gerecht zu werden.“ [20]

Die Anforderungen an barrierefreie Formulare werden in den BITV Bedingungen 10.2, 10.4 und 12.4 formuliert.

Zum einen müssen Sie darauf achten, dass auch bei der Eingabe von Daten in Formularfelder die Geräteunabhängigkeit gewährleistet ist. Beim Durchschalten der Felder mit Hilfe der Tabulator-Taste muss unbedingt die logische Reihenfolge eingehalten werden. Für Adressen wäre dies beispielsweise erst die Straße, gefolgt von der Hausnummer, dann die Postleitzahl und anschließend der Ort. Lässt sich dies nicht allein durch die korrekte Anordnung der Elemente im Quellcode erreichen, so müssen Sie die entsprechende Tab-Reihenfolge per tabindex erzwingen (siehe dazu [Tabindex](#)).



Label

Darüber hinaus muss stets die eindeutige Zuordnung von Eingabefeld und Feld-Beschriftung gegeben sein. Wenn die Anordnung von Formularelementen mit Hilfe von Layouttabellen erreicht wird, kommt es bei der Linearisierung nicht selten zur Trennung von Feldern und den zugehörigen Beschriftungen. Daher gilt auch für Formulare: Verwenden Sie zur Positionierung keine Tabellen, sondern ausschließlich CSS (siehe auch [CSS](#)).

Zur eindeutigen Zuordnung von den sogenannten Etiketten zu ihren Feldern gibt es in HTML das Element `<label>`. Dieses lässt sich über die Vergabe einer id mit dem dazugehörigen Kontrollelement verknüpfen. [21]




Beispiel

Verknüpfung von Formularfeld und Beschriftung


```
<label for="textfeld_name">Name:</label>
<input id="textfeld_name" type="text" />
<br />
<input id="checkbox_mitglied" type="checkbox" value="ja" />
<label for="checkbox_mitglied">Mitglied</label>
```

Checkboxes

Dies bringt außerdem den Vorteil mit sich, dass dadurch auch der Label-Text anklickbar wird. Gerade bei kleinen Checkboxes muss man mit der Maus genau zielen, um diese zu treffen. Die Vergrößerung des anklickbaren Bereichs auf die Beschriftung erleichtert Benutzern mit motorischen Einschränkungen die Aktivierung eines Kontrollkästchens erheblich.

Eingabefelder in Formularen sollten so angelegt werden, dass sie nicht kürzer sind als die mögliche Anzahl an Zeichen, die eingegeben werden können. Wird der Benutzer gezwungen in einem Feld zu scrollen, geht dadurch der Überblick über die bisherige Eingabe verloren. Auf der anderen Seite kann es auch verwirrend sein, wenn vom Nutzer eine kurze Texteingabe erwartet wird, dazu aber ein Feld, das mehrere hundert Zeichen fasst, bereitgestellt wird. Die Feldgröße sollte also visualisieren, welche Länge eine sinnvolle Eingabe hätte.  [MM02]

Pflichtfelder

Für Pflichtfelder gilt das schon in Lerneinheit „LAD - Layout und Design“. Beschriebene: Eine farbliche Unterscheidung reicht als Kennzeichnung allein nicht aus. Für Mussfelder hat sich im Internet die Verwendung des „*“ durchgesetzt. Es sollte vor dem Eingabefeld stehen und idealerweise durch eine fette oder andersfarbige Beschriftung unterstützt werden  [MM02].

Versucht ein Benutzer, ein Formular abzuschicken, in dem nicht alle Pflichtfelder ausgefüllt sind oder die Eingaben nicht der erwarteten Syntax entsprechen, sollte er durch eine aussagekräftige Fehlermeldung darauf aufmerksam gemacht werden. Eine hilfreiche Zusatzfunktion ist die Möglichkeit, den Nutzer durch einen Verweis in der Fehlermeldung direkt zu dem betroffenen Eingabefeld zu leiten und den Cursor in dieses Feld zu setzen.


Darüber hinaus kann es sich als nützlich erweisen, wenn Sie dem Besucher anbieten, ein Formular auszudrucken und per Post an eine angegebene Adresse zu senden.

[20]  www.einfach-fuer-alle.de (Stand: 13.6.2006)

[21]  www.einfach-fuer-alle.de (Stand: 13.6.2006)

10.1 Vorbelegung von Formularfeldern

Vorbelegung

Über den Nutzen von Vorbelegungen für Formularfelder herrscht im Internet Uneinigkeit. Die BITV fordert in Bedingung 10.4: „*Leere Kontrollelemente in Eingabefeldern und Textbereichen sind mit Platzhalterzeichen zu versehen.*“  [BITV02]. Grund hierfür ist die Tatsache, dass einige Zugangsprogramme (zum Beispiel der Textbrowser Lynx) bis heute nicht das `<label>`-Element unterstützen. Bei der Linearisierung kann dadurch die eindeutige Zuordnung von Beschreibungen zu den entsprechenden Eingabefeldern verloren gehen. Eine aussagekräftige Vorbelegung kann hier helfen und Aufschluss über die erwartete Eingabe liefern. „*Zusätzlich kann eine gut getextete Vorbelegung gerade für Computer-Neulinge oder Menschen mit Lernbehinderungen auch in grafischen Browsern eine wichtige Anleitung zur Bedienung eines Formulars sein.*“ [22]

Nachteile der Vorbelegung

Auf der anderen Seite haben Tests mit Screenreadern ergeben, dass blinde Benutzer die Vorbelegung eventuell nicht bemerken und ihre Eingabe tätigen, ohne die Vorbelegung zuvor zu löschen. Als Resultat könnte beispielsweise die Eingabe in einem E-Mail-Feld dann fälschlicherweise „user@domain.deHier bitte die E-Mail-Adresse eingeben“ lauten. [23]

[22]  www.einfach-fuer-alle.de (Stand: 13.6.2006)

[23]  www.einfach-fuer-alle.de (Stand: 13.6.2006)

11 Testing

Tests in der Entwicklungsphase

Während der Entwicklungsphase des HTML-Prototypen ist es unumgänglich, diesen in regelmäßigen Abständen sowie nach größeren Änderungen ausgiebigen Tests zu unterziehen. Dies beinhaltet das Testen der Seiten in unterschiedlichen Browsern, die Validierung des HTML- und CSS-Codes und das Überprüfen der Barrierefreiheit beziehungsweise Zugänglichkeit.



11.1 Browser-Tests

Regelmäßige Überprüfung der korrekten Darstellung

Die regelmäßige Überprüfung der korrekten Darstellung in unterschiedlichen Zugangsprogrammen kann Sie vor späteren unerwarteten Überraschungen bewahren. Denn die Art und Weise, wie Webbrowser HTML- und CSS-Code interpretieren und darstellen, ist bedauerlicherweise oftmals von Programm zu Programm sehr unterschiedlich. So kommt es nicht selten vor, dass eine von Ihnen programmierte Internetseite zum Beispiel im Mozilla Firefox wie gewünscht dargestellt wird, das Layout hingegen im Internet Explorer stark verschoben und in dieser Form gänzlich unbrauchbar erscheint. Damit sich solche Kompatibilitätsfehler nicht weiter fortpflanzen können, ist es entscheidend, sie früh zu erkennen und zu beseitigen.

Grafische Browser

Zahlen zur Nutzung der verschiedenen Browser gibt es im Internet einige. Sie stellen jedoch lediglich eine nicht repräsentative Stichprobe dar. Abweichungen können von der Erhebungsmethode, regionalen Unterschieden und auch die verwendeten Webtechnologien verursacht werden. Schauen Sie sich daher immer mehrere Quellen an und versuchen Sie Trends festzustellen.

<http://www.browser-statistik.de>

<http://www.webhits.de>

<http://www.w3counter.com>

<http://gs.statcounter.com>

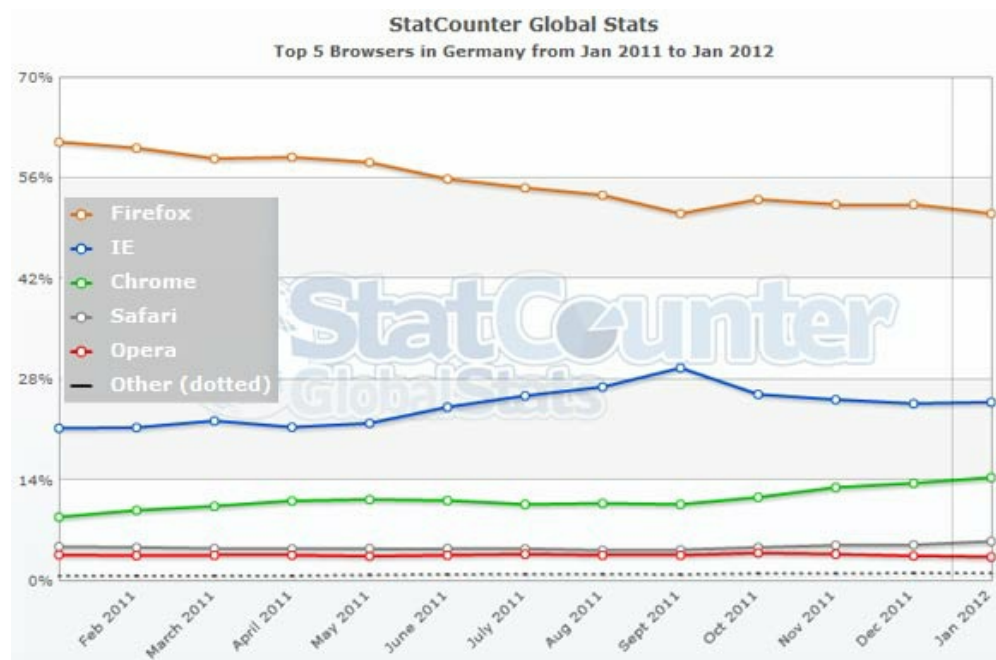


Abb.: Beispielhafte Darstellung der Browserverbreitung der Firma StatCounter

Die Marktanteile unterschiedlicher Webbrowser von Januar 2011 bis Januar 2012






(Quelle der Zahlen: <http://gs.statcounter.com>)

Alle Browser(-versionen) berücksichtigen



Trotz ihrer eher geringen Verbreitung, dürfen Sie jedoch Benutzer von weniger häufig eingesetzten Browsern wie Opera oder älteren Browserversionen wie dem Internet Explorer 7.0 nicht ausschließen. Auch für diese Besucher müssen alle Inhalte einer Website nutzbar sein.

Um die Darstellung und Funktionalität Ihrer Arbeit in einer breiten Auswahl unterschiedlicher Zugangssoftware zu testen, ist es ratsam, sich alle gängigen Browser aus dem Internet herunterzuladen und auf Ihrem System zu installieren. Der parallele Betrieb mehrerer konkurrierender Browser gestaltet sich in der Regel problemlos.



Nachfolgend finden Sie eine Liste der gebräuchlichsten Webbrowser mit den Internetadressen, bei denen Sie diese kostenlos herunterladen können:

-  [Microsoft Internet Explorer](#)
-  [Mozilla Firefox](#)
-  [Chrome](#)
-  [Safari](#)
-  [Opera](#)


Im Hinblick auf die Interpretation von CSS existieren zwischen den verschiedenen Versionen des Microsoft Internet Explorers große Unterschiede. Jedoch ist es nicht so einfach, die Kompatibilität Ihrer Webseiten mit älteren Versionen des Internet Explorers zu testen, da die Installation einer neuen Version in der Regel eine ältere überschreibt.


Inzwischen ist es aber möglich, mehr als eine Version des Internet Explorers auf Ihrem System parallel zu installieren  [An05]. Im  [Evolt Browser Archive](#) finden Sie alle Versionen des Microsoft-Browsers - angefangen bei Version 3.0 - als ZIP-Archive zum Download. Diese müssen Sie lediglich entpacken und die enthaltene Datei iexplore.exe starten.

Text browser

Die korrekte Linearisierbarkeit (siehe  [Linearisierbarkeit](#)) lässt sich mit einem Textbrowser überprüfen. Der wohl bekannteste textbasierte Webbrowser ist zur Zeit  [Lynx](#). Textbrowser interpretieren keine Formatierungen durch Stylesheets und zeigen anstelle von Grafiken, JavaScripts, Flash und anderen Objekten den entsprechenden Alternativtext an. Daher fällt es mit ihnen leicht, die logische Reihenfolge von Navigationen und Inhalten sowie die sinnvolle Strukturierung der Texte zu prüfen.

Screenreader

Blinde oder stark sehbehinderte Menschen können mit der grafischen Darstellung von Webseiten auf dem Bildschirm nichts anfangen. Sie benötigen zur Nutzung des Internets eine Spezialsoftware (den sogenannten Screenreader), welche ihnen den Inhalt des Bildschirms über eine Braillezeile oder per Sprachausgabe zugänglich macht. Einer der bekanntesten Screenreader ist JAWS von der Firma Freedom Scientific. Auf der Website von Freedom Scientific ist eine  [Demo-Version von JAWS zum Download](#) erhältlich, die Sie jeweils für 40 Minuten benutzen können, bevor Sie Ihren Rechner neu starten müssen. Um sich einmal in die Lage eines sehbehinderten Besuchers zu versetzen oder um Ihre Webseiten auf die korrekte Ausgabe durch einen Screenreader zu testen, reicht diese eingeschränkte Version jedoch meist aus.

Eine weitere Möglichkeit, Ihre Internetseiten auf die Lesbarkeit durch blinde Benutzer zu testen, bietet der  [WebFormator](#). Das Programm stellt den Inhalt einer Webseite in einem separaten Textfenster dar. Der Inhalt dieses Fensters ist genau das, was einem sehbehinderten Nutzer über die Sprachausgabe oder Braillezeile in Blinden-Punktschrift zur Verfügung gestellt wird. Es ist also nicht zwangsläufig erforderlich, Zugang zu einer Sprachausgabe-Software oder Braillezeilen Hardware zu haben, um die Kompatibilität Ihrer Seiten mit diesen Hilfsmitteln sicherzustellen.

[24]

[24]  www.webformator.de (Stand: 7.7.2006)

11.2 Tests auf Barrierefreiheit

Automatische und manuelle
Testverfahren

Bei den - im Internet angebotenen - Tests auf Einhaltung der Barrierefreiheit muss zwischen automatischen und manuellen Testverfahren unterschieden werden. Beim automatischen Test müssen Sie dem Online-Testservice lediglich die URL der zu testenden Seite mitteilen. Das System führt alle weiteren Schritte zur Überprüfung der Barrierefreiheit - soweit dies möglich ist - selbstständig durch. Beim manuellen Verfahren müssen Sie die Prüfung selbst vornehmen. Der Testservice unterstützt Sie dabei, indem er Ihnen Anforderungen nennt, anhand derer Sie bewerten müssen, ob die Kriterien der Barrierefreiheit erfüllt werden.



etre
(automatisches Testverfahren)

Der englischsprachige Online-Service ETRE prüft Ihre Webseiten auf Barrierefreiheit gemäß den Richtlinien der WAI. In der Auswertung wird das Ergebnis nach den 3 Prioritätsstufen unterteilt aufgelistet und darin zwischen Fehlern und Warnungen unterschieden. Fehler sind eindeutig feststellbare Verletzungen der Richtlinien, die von Ihnen behoben werden müssen. Warnungen kennzeichnen potentielle Fehlerquellen, die von Ihnen näher geprüft werden sollten.

<http://www.etre.com>

Einige Browser haben integrierte Werkzeuge mit denen sich die aufgerufenen Webseiten auf Accessibility überprüft werden. Dabei wird die Seite an einen Validierungsserver übertragen und ausgewertet. Passwortgeschützte Seiten können nicht validiert werden.

Beim Internet Explorer (9) beispielsweise findet sich das Werkzeug unter „Extras/Entwicklertools (F12)“ und dort unter „Überprüfen/Barrierefreiheit“.

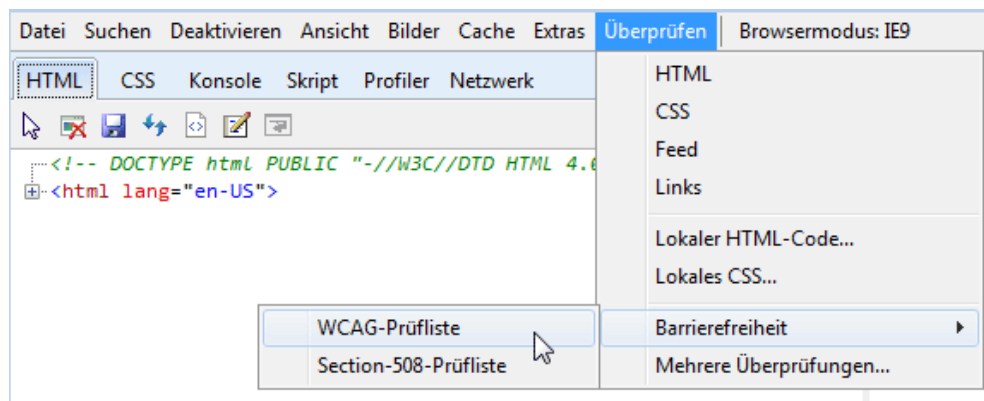


Abb.: Integrierte Validierung im IE9

Nach der Überprüfung einer Seite wird ein Accessibility Report angezeigt, der hier beispielhaft als PDF hinterlegt ist.

[AccessibilityReport.pdf](#) (95 KB)

Manuelle Testverfahren

Bei manuellen Verfahren - beispielsweise die Nutzung eines nicht grafischen Browsers - müssen Sie anhand der angezeigten simulierten Darstellung entscheiden, ob die Seite die Testkriterien erfüllt.

Weitere Informationen zu Testverfahren finden Sie im Internet unter:


[Validierung von Technologien](#)


11.3 Code-Validierung

Syntaktische
Korrektheit

Auf der Website des W3C haben Sie die Möglichkeit, den von Ihnen erstellten Seiten Quellcode auf seine syntaktische Korrektheit zu prüfen.

Mit Hilfe des **Markup Validation Service** lassen sich HTML- und XHTML-Dokumente validieren. Der **CSS-Validator** überprüft CSS-Dateien auf ihre Korrektheit. Nach jedem der Tests werden Sie darüber informiert, ob der geprüfte Code korrekt validiert. Falls dies nicht der Fall ist, listet Ihnen die Seite alle fehlerhaften Code-Segmente auf und gibt Tipps zur Beseitigung der Fehler.

 [Markup Validation Service \(W3C\)](#)

 [CSS-Validator \(W3C\)](#)

11.4 Hilfreiche Zusatztools

Plug-Ins für den Browser

Für den Internet Explorer und für die Browser der Mozilla Familie existieren Zusatztools, die Sie in Form von Plug-Ins kostenlos aus dem Web herunterladen und in Ihren Browser integrieren können. Anschließend stehen Ihnen viele nützliche Erweiterungen zur Verfügung, die sowohl Webdesigner allgemein bei ihrer Arbeit unterstützen können und darüber hinaus Funktionen speziell zur Prüfung der Barrierefreiheit bieten.


Für den Internet Explorer steht die  [Web Accessibility Toolbar](#) von Vision Australia als Download zur Verfügung. Sie bietet Einstellmöglichkeiten in den Bereichen Bildschirmauflösung, Grafiken, CSS und ist an diverse Prüfservices angebunden.



Abb.: Web Accessibility Toolbar

Sie bietet eine Anbindung an verschiedene Testservices sowie viele Einstellmöglichkeiten in den Bereichen Anzeigefenster, CSS, Grafiken, Farbe, Seitenstruktur und einiges mehr.


Benutzer von Mozilla Browsern können diesen durch die  [Web Developer Extension](#) erweitern. Auch sie bietet, wie die Web Accessibility Toolbar, viele hilfreiche Zusatzfunktionen für Webdesigner.



Abb.: Web Developer Extension

Auch sie ist mit mehreren Online-Prüftools verbunden und bietet nützliche Funktionen zur Steuerung von Cookies, CSS, Grafiken, Fenstergrößen und die Anzeige des Quelltexts.

Zusammenfassung

- Bei der Erstellung des HTML-Prototypen müssen Sie im Sinne der Barrierefreiheit auf eine konsequente Trennung von Inhalt und Präsentation achten. Das heißt, dass alle Angaben, die die Darstellung und das Layout einer Webseite betreffen, über CSS-Anweisungen und nicht durch die Verwendung von veralteten HTML-Tags gemacht werden. Diese CSS-Befehle sollten Sie vorzugsweise in eine separate Stylesheet-Datei auslagern, welche dann von allen HTML-Seiten eingebunden wird.
- Dies hat unter anderem den Vorteil, dass der Seiten Quellcode viel übersichtlicher wird und Änderungen am Design der gesamten Website an einer zentralen Stelle vorgenommen werden können. Darüber hinaus ermöglicht ein externes Stylesheet erst eine textorientierte Darstellung, welche für die Barrierefreiheit einer Seite erforderlich ist.
- Im direkten Zusammenhang mit der Forderung, keine Angaben zur Formatierung im HTML-Code zu platzieren, steht das Verbot, Tabellen für Layout-Zwecke zu verwenden. Für einen Screenreader besteht bei einem Layout, das durch die zweckentfremdete Verwendung von Tabellen aufgebaut ist, meist keine Chance, dieses in der logischen Reihenfolge wiederzugeben.
- Für die Gestaltung einer unbeschränkt zugänglichen Seite wird das Layout daher durch den Einsatz von `<div>`-Containern aufgebaut. Diese Behälter lassen sich per CSS beliebig formen und positionieren.
- Einige von der BITV vorgeschriebene Auszeichnungs-Befehle zur Verbesserung der Zugänglichkeit einer Webseite können derzeit leider noch nicht uneingeschränkt empfohlen werden. So bereiten die HTML-Attribute `tabindex` und `accesskey`, welche die Tastaturbedienung des Browsers erleichtern sollen, in vielen Fällen aufgrund einer noch nicht ausgereiften Unterstützung durch die gängigen Webbrowser, mehr Probleme als Nutzen.
- Zur Verbesserung der Verständlichkeit von Texten und um diese durch einen Screenreader vorlesbar zu machen, sind Abkürzungen sowie die Hauptsprache und Sprachwechsel mit dafür vorgesehenen Mitteln zu kennzeichnen.
- In Formularen muss unter anderem die eindeutige Zuordnung von Eingabefeldern und den dazugehörigen Feld-Beschriftungen mit Hilfe des `<label>`-Tags sichergestellt werden.
- Den Abschluss der Prototyping-Phase bildet das ausgiebige Testen aller Webseiten in unterschiedlichen Browsern und die Überprüfung des Quelltextes auf seine Korrektheit. Diverse kostenlose Services im Web bieten die Möglichkeit an, Ihre Internetseiten auf syntaktisch korrekten Code und auf eine barrierefreie Gestaltung zu überprüfen.

Sie sind am Ende dieser Lerneinheit angelangt. Auf der folgenden Seite finden Sie noch die Übungen zur Wissensüberprüfung.

Wissensüberprüfung



Multiple Choice

Übung PRT-01

1. Was bedeutet die Forderung nach einer "Trennung von Inhalt und Design"?

- ☐ Bei der Website-Entwicklung sollte die Erstellung der Inhalte und die Gestaltung des Layouts von unterschiedlichen Abteilungen durchgeführt werden.
- ☐ Der Quellcode mit dem Inhalt und die Formatierungsbeschreibungen einer Webseite sollten in separaten Dateien untergebracht werden.
- ☐ Bei der Entwicklung einer Website sollten Sie den Schwerpunkt entweder auf ein ansprechendes Design legen oder aber auf den Inhalt. Im zweiten Fall kann die Gestaltung eher vernachlässigt werden.

2. Wie lässt sich ein Seitenlayout am einfachsten barrierefrei gestalten?

- ☐ Durch den Einsatz von Stilvorlagen.
- ☐ Durch den Einsatz von Layouttabellen.
- ☐ Durch den Einsatz der Frame-Technik.

3. Welchen Bereichen des CSS-Boxmodells können Sie eine Hintergrundfarbe zuweisen?

- ☐ Dem Außenabstand und dem Innenabstand.
- ☐ Dem Inhalt und dem Rahmen.
- ☐ Allen vier Bereichen.

4. Wie lässt sich eine Bereichsüberschrift, die von einem Screenreader vorgelesen werden soll, am besten in der grafischen Darstellung verbergen?

- ☐ Durch den CSS-Befehl `display:none`;
- ☐ Durch den CSS-Befehl `visibility:hidden`;
- ☐ Durch eine CSS-Positionierung außerhalb des sichtbaren Bereichs

5. Welche Aussage ist falsch?

- ☐ Die Vergabe von `tabindex`-Nummern ist nicht erforderlich, wenn die Standard Tab-Reihenfolge einer Webseite bereits eine sinnvolle Reihenfolge ergibt.
- ☐ Je nach Art des Webbrowsers sind viele Tastenkürzel schon belegt, so dass eine Definition von Access-Keys wenig sinnvoll ist.
- ☐ Die Bereitstellung einer Sitemap kann für die Besucher eines Internet-Angebots sehr hilfreich sein, ist aber für die Barrierefreiheit einer Website nicht zwingend erforderlich.

6. Worauf müssen Sie bei der Gestaltung barrierefreier Formulare besonders achten?

- ☐ Beim Durchschalten der Felder mit Hilfe der Tabulator-Taste muss unbedingt die logische Eingabereihenfolge eingehalten werden.
- ☐ Zur eindeutigen Zuordnung von einem Eingabefeldern zu der dazugehörigen Beschriftung muss unbedingt das HTML-Element verwendet werden.
- ☐ Sowohl auf eine logische Eingabereihenfolge als auch auf eine eindeutige Zuordnung der Feldbeschriftungen muss geachtet werden.