

Checkliste: Richtlinien zur Barrierefreiheit bei eLearning-Projekten in Kooperation mit den neuen Medien und Lerntechnologien (Arbeitsversion) – 12. 10. 2006

Zentraler Informatikdienst
Neue Medien und Lerntechnologien
DI (FH) Ortrun Gröbinger

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Einleitung	1
Empfehlungen für den Webbereich	2
Verständliche Dokumentationsgestaltung	2
Klare Navigationsmechanismen	3
Natürliche Sprache	5
Tabellen	6
Alternativen zu Audio und visuellen Inhalten	7
Farbe	10
Markup	11
Stylesheets	13
Benutzerkontrolle über zeitgesteuerte Abläufe	13
Technologien und Richtlinien von W3C	14
Quellen	14

Einleitung

Was sind Barrieren im Internet? Diese Frage lässt sich nicht direkt beantworten, da Barrieren für jeden etwas anderes bedeuten. Müssen in erster Linie die Bedürfnisse von Menschen mit Behinderungen wie Sehschwächen, Hörschäden, eingeschränkter Beweglichkeit usw. berücksichtigt werden, gibt es eine breite Benutzergruppe, die z.B. nur über eine langsame Internetanbindung oder gering auflösende Bildschirme verfügt. Man denke weiters an den aktuellen Trend Mobile Endgeräte auch im Lernbereich einzusetzen. Die Individualisierung der Lernumgebung und Lernmaterialien werden mittlerweile als ein wichtiger Aspekt für gutes Lernen angesehen. Das führt zu sehr hohen Ansprüchen an die Materialentwicklung, und zwar sowohl auf inhaltlicher, als auch auf technischer Ebene.

Als Materialproduzent bzw. Materialproduzentin befinden Sie sich heutzutage in einer Zwickmühle. Einerseits gibt es eine geradezu gigantische Auswahl an Möglichkeiten um Inhalte am Computer, insbesondere für das Internet, bunt, vertont, mit Animationen, Grafiken, Videos, Textgestaltungen usw. lernfördernd, motivierend und künstlerisch aufzuarbeiten.

Andererseits müssen bestimmte Richtlinien beachtet werden um Barrierefreiheit zu wahren, nicht zuletzt aufgrund des E-Government-Gesetzes welches mit 1. Jänner 2008 in Österreich in Kraft treten wird. In diesem Gesetz ist unter anderem verankert, dass alle Behördlichen und Öffentlichen Seiten, für alle Menschen zugänglich sein müssen. In diese Definition fallen sowohl Ämter, als z.B. auch Universitäten und bis zu einem gewissen Grad auch die dort angebotenen Lernmaterialien.

Wie barrierearm sich eine Universität präsentieren möchte, welchen Standard sie für sich realisieren möchte, ist eine Profilscheidung. Die Universität Innsbruck ist daran interessiert, möglichst barrierearm aufzutreten. Aus diesem Grund wurde nach Richtlinien, Vereinbarungen und Kompromissen gesucht, die sich auf den Bereich der Neuen Medien wie folgt auswirken werden:

Für den Webbereich wurden zehn Bereiche von den Mitarbeitern der „Neuen Medien und Lerntechnologien“ ausgewählt. Zu diesen Bereichen wurden Richtlinien verfasst. Diese treten in Zukunft bei geförderten eLearning-Projekten, die in Zusammenarbeit mit der Abteilung Neue Medien und Lerntechnologien erstellt werden, in Kraft.

Diese Richtlinien decken in erster Linie die Basisvoraussetzungen für Barrierefreiheit ab, sind aber bei weitem nicht vollständig – die vollständigen Richtlinien entnehmen Sie bitte z. B. den W3C-Richtlinien. Eine selbstständige, weitere Verbesserung der Projektmitarbeiter am Produkt ist gewünscht. In diesem Zusammenhang beraten Sie die Mitarbeiter der Abteilung „Neue Medien und Lerntechnologien“(ZID) in Kooperation mit den Behindertenbeauftragten der Universität Innsbruck gerne.

Empfehlungen für den Webbereich

1. Verständliche Dokumentationsgestaltung
2. Klare Navigationsmechanismen
3. natürliche Sprache
4. Tabellen
5. Alternativen zu Audio und visuellen Inhalten
6. Farbe
7. Markup
8. Stylesheets
9. Benutzerkontrolle über zeitgesteuerte Abläufe
10. Technologien und Richtlinien von W3C

Bei Problemen wenden Sie sich jederzeit an das Team der Neuen Medien und Lerntechnologien (<http://www.uibk.ac.at/zid/abteilungen/lt/personen>).

Verständliche Dokumentationsgestaltung

1.1 Verwenden Sie für die Inhalte einer Seite die klarste und einfachste Sprache, die angemessen ist.

Die gewählte Sprache muss der Zielgruppe entsprechen.

1.2 Ergänzen Sie Text mit grafischen oder audiovisuellen Präsentationen, wo dies das Verständnis der Seite erleichtert.

1.3 Verwenden Sie einen Präsentationsstil, der über Seiten hinweg konsistent ist.

Achten Sie beispielsweise darauf:

- Immer dieselben Aufzählungssymbole für dieselben Kategorien zu verwenden.
- Farbkodierungen, Schriftarten, Schriftgrößen,... je nach Einsatz beizubehalten.
- Überschriften immer nach demselben Prinzip zu kennzeichnen.
- Navigationselemente immer auf der gleichen Stelle am Bildschirm zu platzieren.

Klare Navigationsmechanismen

2.1 Identifizieren Sie das Ziel jedes Links auf klare Weise (d.h. vermeiden Sie z.B. ständig links mit „hier klicken“, „weiter“, „mehr“ oder „next“ zu setzen.

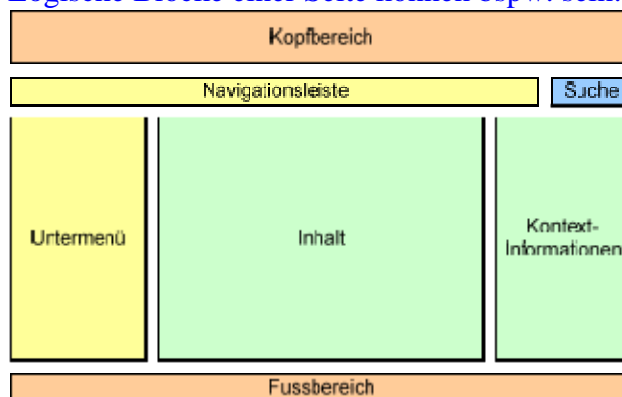
Vor allem blinde Anwender verschaffen sich gerne einen raschen Überblick über eine ihnen unbekannt Seite, indem Sie mit der Tabulatortaste von Link zu Link springen und im ersten Schritt nur diese lesen. Wenn sich auf einer Homepage ausschließlich Links befinden, die z.B. mit „weiter“ beschriftet wurden, die jedoch nicht sagen, welche Information sich hinter dem Link verbirgt, bedeutet das einen vielfachen Arbeitsaufwand zur Erschließung der Seite für blinde Benutzer.

2.2 Verwenden Sie klar strukturierte Navigationen.

Problem: Nicht jeder kann auf einer Seite schnell navigieren. Manche Benutzer müssen sehr viel überlesen, um sich auf der Seite zu Recht zu finden (z.B. wenn die Navigationsleiste sehr viele Einträge hat).

In solchen Fällen müssen Mechanismen angeboten werden, die die Navigation erleichtern. Eine oft praktizierte Möglichkeit ist das Vergeben von internen Accesskeys (0-9), die zu den verschiedenen logischen Blöcken einer Seite führen. Zusätzlich sollte jeder dieser Blöcke Metainformation beinhalten.

Logische Blöcke einer Seite können bspw. sein:



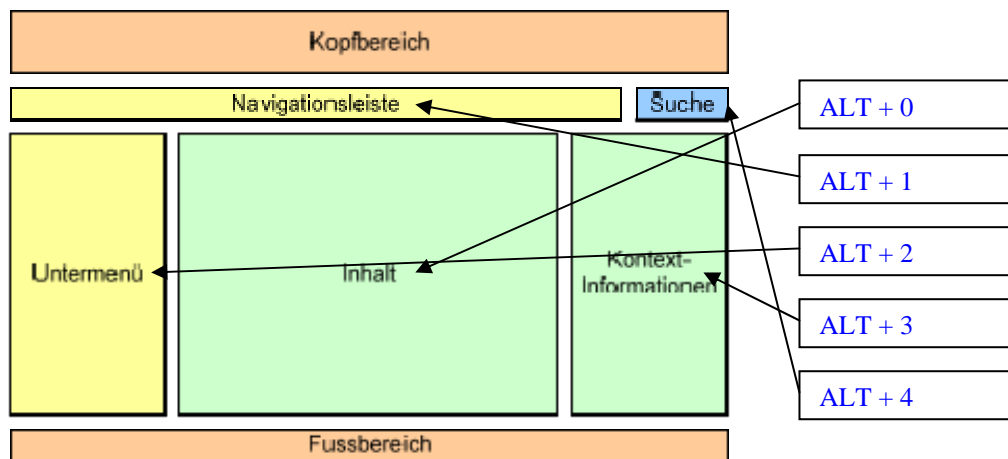
Grafik: 7 Logische Blöcke einer Website (mehr als 7 zu identifizieren macht üblicherweise keinen Sinn)

```
<body>
<ul>
  <li><a href="#content" accesskey="0">Zum Inhalt (ALT+0)</a> |
</li>
```

```

<li><a href="#navbar" accesskey="1">Zum Inhalt (ALT+1)</a> | </li>
...
</ul>
...
<div class="navbar"> <a name="navbar">Navigationsleiste:</a> ...
  <hr />
</div>
<div class="content"> <a name="content">Inhalt:</a> ...
  <hr />
</div>
...
</body>

```



Grafik: Übliche Vergabe der Accesskeys (es existieren bislang keine verbindlichen Richtlinien für die Reihenfolge. Die hier angegebene ist jedoch durchaus geläufig)

2.2.1 Stellen Sie Navigationsleisten zur Verfügung

Manche Benutzer haben Probleme damit, geschachtelten Untermenüs zu verstehen (z.B. lesen die meisten Screenreader die Menüpunkte alle der Reihe nach vor, ohne darauf Rücksicht zu nehmen, ob es sich um ober- oder untergeordnete Punkte handelt. Weiters werden im Normalfall alle Menüpunkte vorgelesen, auch solche, die zugeklappt sind – d.h. für einen sehenden Benutzer nicht sichtbar sind. Bei sehr langen Menüs ergeben sich dadurch Schwierigkeiten).

Eine Lösung wäre der Einsatz von geschachtelten Listen. Bei dieser Verschachtelung werden zusätzliche Informationen über die Hierarchieebene angeboten.

```

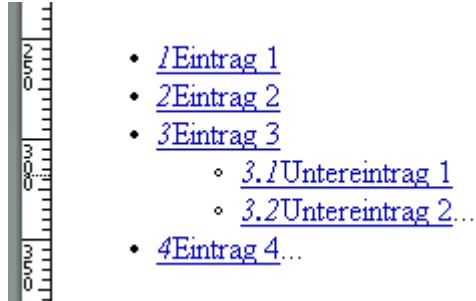
<body>
<map name="menu" title="Menü">
  <ul>
    <li><a href="..."><dfn>1</dfn>Eintrag 1</a></li>
    <li><a href="..."><dfn>2</dfn>Eintrag 2</a></li>
    <li><a href="..."><dfn>3</dfn>Eintrag 3</a>
      <ul>
        <li><a href="..."><dfn>3.1</dfn>Untereintrag 1</a></li>
        <li><a href="..."><dfn>3.2</dfn>Untereintrag 2</a></li>
        ...
      </ul>
    </li>
  </ul>
</map>

```

```

    <li><a href="..."><dfn>4</dfn>Eintrag 4</a></li>
    ...
  </ul>
</map>
</body>

```



Grafik: Darstellung des Codebeispiels. Das Erscheinungsbild dieser Liste kann mit CSS beinahe beliebig angepasst werden.

2.2.2 Gruppieren Sie verwandte Links

Es sollte sich schon in der Navigation zeigen, welchen zusammenhängende Themenblöcke auf einer Seite behandelt werden.

Natürliche Sprache

3.1 Geben Sie zu Beginn des Dokuments dessen natürliche Sprache an.

D.h. geben Sie bitte die Sprache bekannt, in der der Großteil der Texte auf der jeweiligen Website verfasst wurde.

3.2. Kennzeichnen Sie Änderungen in der natürlichen Sprache in HTML mit dem „lang“-Attribut und in XML mit „xml:lang“. (Anmerkung: lang ist ein Universalattribut. D.h. das es in mehreren HTML-Elementen vorkommen darf. „lang“ kann in allen Attributen eingesetzt werden, außer: applet, base, basefont, br, frame, frameset, hr, iframe, meta, param, script)

Beispiel HTML:

```

<body>
...
<p> Heute unterhalten wir uns über <ACRONYM lang="eng"
TITLE="englisch: Compact Disc - Read Only Memory">CD-ROM</ACRONYM>.
</p>
...
</body>

```

Beispiel XML:

```

<body>
...
<p> Hier können Sie mehr zum Thema <span lang="en"
xml:lang="en">Copyright </span> lesen. </p>
...
</body>

```

3.3 Erklären Sie jede Abkürzung und jedes Akronym an der Stelle ihres ersten Auftretens (in HTML mit „title“-Attribut und den Elementen ABBR [für Abkürzungen] und ACRONYM).

Beispiel ABBR:

```
<body>
...
<p> Gestern kam es zu einem verheerenden Unfall, in welchem zwei
<ABBR TITLE="Personenkraftwagen">PKW </ABBR>und <ABBR
TITLE="Lastkraftwagen">LKW</ABBR> verwickelt waren. </p>
...
</body>
```

Beispiel ACRONYM:

```
<body>
...
<p> Heute unterhalten wir uns über <ACRONYM lang="eng"
TITLE="englisch: Compact Disc - Read Only Memory">CD-ROM</ACRONYM>.
</p>
...
</body>
```

Tabellen

4.1 Tabellen sind nicht zu Layout-Zwecken gedacht. Verwenden Sie Tabellen daher ausschließlich zur Übermittlung von Daten, deren Präsentation in Tabellenform zweckdienlich ist.

4.1.1 Kennzeichnen Sie bei Datentabellen Zeilen- und Spaltenüberschriften (in HTML mit TD für Datenzellen, TH für Überschriften).

4.1.2 Stellen Sie eine Zusammenfassung zu jeder größeren Tabelle bereit (Attribut: summary).

Einfache Tabelle:

```
<table border="1" summary="Accesskeys der Seite">
  <caption>Accesskeys in der Navigation:</caption>

  <thead>
    <tr>
      <th id="t">Taste</th>
      <th id="b">Beschreibung</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td headers="t">ALT + 0</td>
      <td headers="b">Inhalt</td>
    </tr>
    <tr>
      <td headers="t">ALT + 1
      <td headers="b">Hauptnavigation</td>
    </tr>
    [...]
  </tbody>
</table>
```

```
</tbody>
</table>
```

...

Accesskeys in der Navigation:

Taste	Beschreibung
ALT + 0	Inhalt
ALT + 1	Hauptnavigation
ALT + 2	Untermenü
ALT + 4	Kontextinformation
ALT + 5	Suche

...

Grafik: Zeigt einen Screenshot des ausgeführten, s vorangegangenen Codebeispiels

4.1.3 Gruppieren Sie zueinander gehörende Daten (in HTML mit THEAD, TFOOT und TBODY, COL und COLGROUP).

4.1.4 Vermeiden Sie leere GIFs (Bilder) zu Platzhalterzwecken. Sollte es unbedingt notwendig sein ein leeres GIF einzusetzen, verwenden Sie bitte folgende Beschriftung:

```
<p><p>
```

Alternativen zu Audio und visuellen Inhalten

5.1 Stellen Sie für jedes Nicht-Text-Element ein Text-Äquivalent zur Verfügung; d.h. beschreiben Sie jedes Bild, jedes Video, jede Audioaufnahme, selbstablaufende Präsentationen etc, in textlicher Form.

5.1.1 Verwenden Sie „alt“ für die IMG-, INPUT- und APPLET-Element oder stellen Sie ein Text-Äquivalent im Inhalt des OBJECT- und APPLET-Element bereit.

Beispiel: Eingebundene Grafik

```
<p><p>
```

Beispiel: Formular mit Input-Attribut

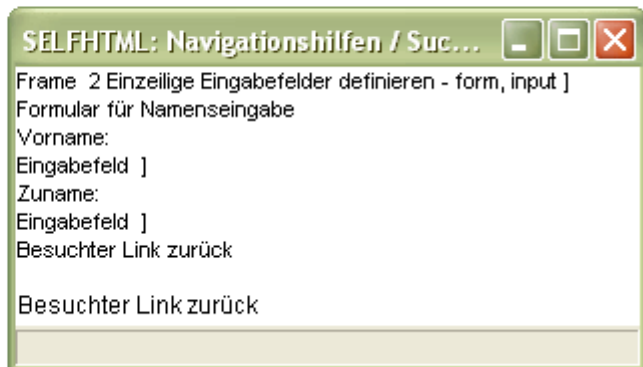
```
<body>
<h1>Formular f&uuml;r Namenseingabe</h1>
<form action="input_text.htm">
  <p>Vorname:<br><input name="vorname" type="text" size="30"
maxlength="30"></p>
  <p>Zuname:<br><input name="zuname" type="text" size="30"
maxlength="40"></p>
</form>
</body>
```

Formular für Namenseingabe

Vorname:

Zuname:

[zurück](#)



Bei diesem Beispiel sehen Sie im rechten Bereich die Ausgabe eines Screenreaders. Da die Formatierung der Formularfelder korrekt vorgenommen wurde, kann ein eindeutiger Zusammenhang zwischen der Beschriftung und dem jeweiligen Eingabefeld hergestellt werden.

- 5.1.2 Stellen Sie für komplexen Inhalt, bei dem das „alt“-Tag nicht genügt, zusätzliche Beschreibungen z.B. mit „longdesc“ im IMG und FRAME, einen Link innerhalb eines OBJECT-Elements, oder einen Beschreibungs-Link.

Bei Bildern (IMG):

Wenn die Möglichkeit, mit dem alt-Attribut einen gewöhnlichen Alternativtext für eine Grafik zu notieren, nicht genügt, kann seit HTML 4.0 auch zu einer anderen Stelle oder Datei verweisen, wo sich nähere Textinformation zu der Grafik befindet. Der Browser könnte beispielsweise nur dann einen Verweis zu dieser Stelle anbieten, wenn die Grafik nicht anzeigbar ist.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    „http://www.w3.org/TR/html4/strict.dtd“>
<html>
  <head>
    <title>Langbeschreibung f&uuml;r Grafiken</title>
  </head>
  <body>
    <h1>Alte Zeiten</h1>
    <p><img scr="altezeiten.jpg" width=271 height="265" alt="alte
Zeiten" longdesc="altezeiten.htm"></p>
  </body>
</html>
```


Adresse http://de.selfhtml.org/html/grafiken/anzeige/img_longdesc.htm Wechseln zu Links Google iPoint

Alte Zeiten



Langbeschreibung für Grafiken - longdesc WebFormator (1)

Seite Langbeschreibung für Grafiken - longdesc http://de.selfhtml.org/html/grafiken/anzeige/img_longdesc.htm
 Alte Zeiten
 Grafik alte Zeiten]
 Besuchter Link zurück

[zurück](#)

Bei Frames (FRAME):

Mit dem Attribut longdesc in einem <frame>-TAG können Sie auf eine Stelle verweisen, wo der Inhalt dieses Frame-Fensters in einer längeren Beschreibung zusammengefasst wird. Erlaubt ist ein URI, d.h. bei der Angabe gelten die Regeln zum Referenzieren in HTML. Dieses Attribut kommt bei Browsern zum Einsatz, die Frames zwar erkennen, aber bei denen die Anzeige von Frames nicht möglich ist, wie z.B. bei akustische Ausgabesystemen.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>Langbeschreibung für Frame-Fenster</title>
</head>
<frameset cols="250,*">
  <frame src="verweise.htm" name="Navigation">
  <frame src="startseite.htm" name="Daten"
  longdesc="startseite_extra.htm">
  <noframes>
  <body>
    <p>Alternativ-Inhalt (Zusammenfassung, Sitemap ...)</p>
  </body>
  </noframes>
</frameset>
</html>

```

- 5.1.3 Vermeiden Sie wenn möglich Imagemaps. Sollten Sie Imagemaps einsetzen, verwenden Sie entweder das „alt“-Attribut bei AREA oder das MAP-Element mit A-Elementen (und zusätzlichem Text) als Inhalt; d.h. stellen Sie redundante Textlinks zur Verfügung.

Beispiel siehe: http://de.selfhtml.org/html/grafiken/verweis_sensitive.htm#definieren
(Verweis-sensitiven Grafiken definieren)

- 5.2 Beim unvermeidbaren Einsatz von Platzhalterbildern, oder auch bei dekorativen Elementen, verwenden Sie ein leeres alt-Attribut: alt="";

```
<p><p>
```

Farbe

- 6.1 Sorgen Sie dafür, dass alle Ihre Informationen auch gänzlich ohne Farbe verständlich dargestellt werden (schwarz-weiß, keine Graustufen).

- 6.2 Achten Sie auf einen ausreichenden Kontrast bei Vorder- und Hintergrundfarbe.

„Von einer guten Sichtbarkeit zweier Farben kann dann ausgegangen werden, wenn der Helligkeitsunterschied und die Farbdifferenz zwischen den beiden Farben größer ist als ein bestimmter Grenzwert. Der von W3C vorgeschlagene Grenzwert ist > 125 für die Helligkeit und > 500 für die Farbdifferenz“.

Es stehen einige Programme zur Verfügung, mit welchen sich diese Kontraste berechnen lassen.

Formel für die Helligkeit von Farben:

$$((\text{Rot-Wert} \times 299) + (\text{Grün-Wert} \times 587) + (\text{Blau-Wert} \times 114)) / 1000$$

Der Unterschied zwischen der Helligkeit des Hintergrundes und des Vordergrundes sollte größer als 125 sein.

Formel für den Farbunterschied:

$$(\text{Maximum}(\text{Rot-Wert1}, \text{Rot-Wert2}) - \text{Minimum}(\text{Rot-Wert1}, \text{Rot-Wert2})) + \\ (\text{Maximum}(\text{Grün-Wert1}, \text{Grün-Wert2}) - \text{Minimum}(\text{Grün-Wert1}, \text{Grün-Wert2})) + \\ (\text{Maximum}(\text{Blau-Wert1}, \text{Blau-Wert2}) - \text{Minimum}(\text{Blau-Wert1}, \text{Blau-Wert2}))$$

Die Differenz zwischen Hintergrundfarbe und Vordergrundfarbe sollte größer als 500 sein.

Analysiert ob die Kontraste zwischen Hintergrund und Text stark genug sind.

<http://juicystudio.com/services/colourcontrast.php>

- 6.3. Denken Sie bei der Gestaltung an Personen mit Farbfehlsichtigkeiten (wie z.B. Rot-Grün Farbenblindheit). Informationen wie „Drücken Sie den grünen Button“ sind für diese Benutzergruppe nicht brauchbar.

Vischeck simuliert, wie Bilder für Farbenblinde dargestellt werden.

<http://www.vischeck.com/>

Markup

7.1. Erstellen Sie Dokumente, welche den formalen Grammatiken der Programmiersprache entsprechen.

7.1.1 Verwenden Sie eine Dokumententype-Deklaration am Anfang Ihres Dokuments.

7.1.2 Verwenden Sie ausschließlich W3C-konforme TAGS (HTML-TAGs, XHTML-TAGs, etc.)

7.1.3 Zeichnen Sie alle Teile Ihrer Web-Dokumente mit entsprechenden TAGS aus.

`<head>`, `<body>`, `<h1>`, `<p>`, ...

7.1.4 Verwenden Sie Überschriften-Elemente (h1, h2,...), um die Struktur eines Dokuments darzustellen.

7.1.5 Verwenden Sie Markup für Listen und Listenelemente (OL-, UL- und DL-Listen).

`` steht für sortierte Listen (ol = odered list). `` wird in Nummerierungen umbewandelt:

```
<h1>Sortierte Liste</h1>
<ol>
  <li>auwachen</li>
  <li>aufstehen</li>
  <li>waschen</li>
  <li>anziehen</li>
  <li>frühstücken</li>
  <li>Zähneputzen</li>
  <li>frisieren</li>
</ol>
```

`` leitet unsortierte Listen (ul = unordered list) ein. Mit `` beginnt ein neuer Punkt innerhalb der Liste (li = list item).

Listen können auch verschachtelt werden.

```
<h1>Unsortierte Liste</h1>
<ul>
  <li>Großkatzen
    <ul>
      <li>Löwe</li>
      <li>Tiger</li>
      <li>Leopard</li>
    </ul>
  </li>
  <li>Paarhufer
    <ul>
      <li>Schweineartige</li>
      <li>Schwielensohler</li>
      <li>Wiederkäuer</li>
    </ul>
  </li>
</ul>
</body>
```

Sortierte Liste

1. auwachen
2. aufstehen
3. waschen
4. anziehen
5. frühstücken
6. Zähneputzen
7. frisieren

Unsortierte Liste

- Großkatzen
 - ◊ Löwe
 - ◊ Tiger
 - ◊ Leopard
- Paarhufer
 - ◊ Schweineartige
 - ◊ Schwielensohler
 - ◊ Wiederkäuer

Definitionslisten <dl> sind für glossare gedacht.

<dl> leitet eine Definitionsliste ein (dl = definition list)

<dt> leitet einen zu definierenden Ausdruck ein (dt = definition (list) item)

<dd> leitet eine Definition eines Ausdrucks ein (dd = definition (list) definition)

```
<h1>Abkürzungen</h1>
<dl>
  <dt>g</dt>
  <dd>grin</dd>
  <dt>fg</dt>
  <dd>fed grin</dd>
  <dt>eg</dt>
  <dd>evil grin</dd>
</dl>
```

Abkürzungen

```
g
  grin
fg
  fed grin
eg
  evil grin
```

7.1.6 Wenn eine angemessene Markup-Sprache existiert, verwenden Sie Markup anstelle von Bildern, um Informationen darzustellen (z.B. MathML).

7.1.7 Verwenden Sie Markup für Zitate (in HTML Q und BLOCKQUOTE, um kürzere und längere Zitate zu kennzeichnen).

```
<body>
<h1>Ernest Hemingway</h1>

<p>Hemingway war nicht nur Schriftsteller, sondern auch Reporter und
Kriegsberichterstatter.</p>

<blockquote>
  <p>As you get older it is harder to have heroes, but it is sort of
necessary. </p>
</blockquote>

</body>
```

Stylesheets

8.1 Bauen Sie Dokumente so auf, dass sie ohne Stylesheet gelesen werden können.

Hiermit ist gemeint, dass die wichtigsten Informationen einer Seite auch dann korrekt gelesen werden können, wenn dem Benutzer keinerlei Layout zur Verfügung steht. D.h. dass darauf geachtet werden muss, dass wichtige Informationen

- nicht ausschließlich über die Formatierung vermittelt werden und
- Informationen auch ohne Style-Angaben in der richtigen Reihenfolge übermittelt werden.

8.2. Trennen Sie sämtliche layoutspezifischen Angaben in Websites wie z.B.: Hintergrund- und/oder Schriftfarben, Schriftarten, Schriftgrößen, Bilder, Roll-Over-Effekte etc. vom reinen Homepagecode [(X)HTML], und setzen Sie dafür CSS ein.

8.3 Verwenden Sie relative anstatt absolute Einheiten bei Größenangaben und Platzierungen (anstelle von <px>).

Benutzerkontrolle über zeitgesteuerte Abläufe

9.1 Geben Sie den Benutzern immer die Möglichkeit, Abläufe (z.B. von Videos, Präsentationen) selbst zu steuern.

9.1.1 Vermeiden Sie Bildschirmflackern, blinkende Inhalte, automatische Seitenrefreshs, automatische Seitenweiterleitungen und bewegte Seiten, bis Benutzeragenten die Kontrolle darüber erlauben.

Anmerkung: Benutzeragenten sind Programme, mit denen der Benutzer bestimmte Abläufe steuern kann. Dazu gehören z.B. Browser für HTML oder XHTML, Bildbetrachtungsprogramme oder spezifische Technologien für Behinderte wie Screenreader, Braillezeile, Programme die Bildschirminhalte vergrößert darstellen, etc.

Technologien und Richtlinien von W3C

10.1 Verwenden Sie W3C-Technologien, wenn diese verfügbar und angemessen sind, und die dem aktuellen Stand der Technik entsprechen.

W3C-Technologien sind z.B. HTML, XHTML, XML, SMIL, SVG, CSS, RDF,...

Viele Nicht-W3C-Formate (z.B. PDF, Shockwave,...) können mit Standard-Benutzeragenten (einschließlich assistiver Technologien) nicht ohne weiteres verarbeitet (betrachtet) werden. Die Vermeidung proprietärer Technologien (proprietäre Elemente, Attribute,...) wird in der Tendenz Seiten die Zugänglichkeit von Seiten für Menschen mit Einschränkungen verbessern. Wenn nicht zugängliche proprietäre Technologien (egal ob proprietär oder nicht) verwendet werden müssen, müssen äquivalente zugängliche Seiten bereitgestellt werden.

Quellen

Binder, Urs: Barrierefreiheit kostet nichts, InfoWeek.ch Nr. 10, 17. 05. 2005, S. 40 – 42, letzter Zugriff: 28. August 2006

<http://www.namics.com/wissen/artikel-interviews/artikel/barrierefreiheit-kostet-nichts.html#275>

W3C Recommendation, 5. Mai 1999, letzter Zugriff: 28. August 2006

<http://www.w3c.de/Trans/WAI/webinhalt.html#Guidelines>

Aktion Mensch und Stiftung Digitale Chancen: Kriterien zum BIENE-Award 2006, letzter Zugriff: 28. August 2006

<http://www.biene-award.de/award/kriterien/>

Nussbaum, Gerhard: Vortrag im Rahmen des eLearning Tags „eLearning an der LFUI – Rückblick und Ausblick“: Barrierefreie Web-Gestaltung, 28. September 2006

HTML-Referenz: SELFHTML ,letzter Zugriff: 10. Okt. 2006

<http://de.selfhtml.org>