

BITV für Alle

Seit dem Inkrafttreten des [Gesetzes zur Gleichstellung behinderter Menschen](#) ist Barrierefreiheit nicht mehr nur eine nette Geste, sondern für öffentliche Anbieter Pflicht. Ab dem 1. August 2002 müssen alle Seiten, die von Bundesbehörden ins Netz gestellt werden, bestimmten Kriterien an die Barrierefreiheit genügen. Entsprechende Gesetze und Verordnungen auf Landes- und kommunaler Ebene sind in Vorbereitung und werden sich an die Vorgaben der BITV anlehnen.

[Diese Verordnung](#) hat weitreichende Folgen für alle, die mit der Erstellung staatlicher Internetpräsenzen befasst sind, unabhängig davon, ob es interne Projekte sind, oder ob externe Agenturen beauftragt werden. Obwohl es bereits einige [herausragende Beispiele](#) gibt, besteht noch viel Nachholbedarf bei den Webentwicklern und Projektverantwortlichen.

Daher werden wir hier in einer 14-tägigen Serie die einzelnen Anforderungen der BITV in getrennten Artikeln detaillieren und erklären, wie Sie Ihre Seiten auf BITV-Konformität testen. Zusätzlich gibt es zu jeder Bedingung noch ein leicht nachvollziehbares Praxisbeispiel.

Einige grundlegende Definitionen

Wenn wir in dieser Serie von optionalen Formaten sprechen, so sind damit alle Inhalte einer Webseite gemeint, die nicht HTML oder Text sind. Dazu zählen so offensichtliche Dinge wie JavaScript, Java, Flash, Videos und Sounds, aber auch Bilder und Grafiken, CSS, Frames und hartcodierte Formatierungen im HTML selbst. Letztlich alles, was sich in irgendeinem Browser abschalten lässt, ist somit im Sinne der BITV und damit für die folgenden Tests ein optionales Format.

Die häufige Nennung bestimmter Browser stellt natürlich keine Wertung dar. Wenn sie einen anderen Browser in Ihrem täglichen Umgang mit dem Web einsetzen, so können Sie dies auch gerne weiter tun. Wir beschränken uns hier nur auf eine bestimmte Anzahl Browser, weil diese über Features verfügen, welche die beschriebenen Tests vereinfachen oder überhaupt erst ermöglichen.

Die BITV ist wortgemäß nur für den Zuständigkeitsbereich des Bundes gültig. Es ist aber zu erwarten, dass die Vorgaben sinngemäß auch in die Landesgleichstellungsgesetze übernommen und somit auch auf Länder und Kommunen anwendbar sein werden.

Die einzelnen Bedingungen der BITV sind in den folgenden Artikeln nicht entsprechend Ihrer fortlaufenden Nummerierung aufgelistet, sondern je nach Sinnzusammenhang mit anderen verwandten Bedingungen zusammengefasst. So kann es vorkommen, dass eine Bedingung aus 13.6 weiter vorne bei 12.3 auftaucht. Die z.Zt. in der Entwicklung befindlichen [WCAG 2](#) gehen einen ähnlichen Weg. Hier werden die Vorgaben aus der Version 1 weiter zusammengefasst, so dass es nur noch 5 Kategorien geben wird.



Ihre Einkaufsliste:

Die in den folgenden Tagen durchgeführten Tests funktionieren leider nicht mit jedem Browser. Nur Browser, die auf einer modernen Codebasis beruhen und z.B. das W3C DOM vollständig unterstützen, können überhaupt die notwendigen Aktionen in den Seiten durchführen. Daher haben wir uns weitestgehend auf Mozilla und Opera beschränkt. Die folgenden Browser und Tools sind entweder kostenlos, oder es gibt zumindest, wie bei den Screenreadern, eine für begrenzte Zeit lauffähige Testversion:

Browser:

- [Mozilla](#)
- [Opera](#)

Browser-Erweiterungen:

- [Checky für Mozilla](#)
- [Preferences Toolbar für Mozilla](#)
- [PNH Developer Toolbar](#)
- [ToggleCSS-Favelet](#)
- [Context Menu Extension](#)

Screenreader bzw. Sprachbrowser (optional):

- [Screenreader JAWS](#)
- [Testversion des Home Page Reader](#)

Lizenzbestimmungen für Open Content

Copyright 'Einfach für alle' © 2003 Aktion Mensch e.V. und die jeweiligen Autoren.

Dieses Werk kann gemäß den Bestimmungen der Lizenz für die freie Nutzung unveränderter Inhalte genutzt werden, Sie finden diese Lizenzbestimmungen unter <http://www.einfach-fuer-alle.de/lizenz/>.

Die Verordnung im Detail:

Tag 1: Alternativen	4
Am ersten Tag der BITV-Serie geht es um einige grundlegende Dinge, die schon immer zum 'Guten Ton' einer Website gehörten.	
Tag 2: Farben	8
Zirka zehn Prozent Ihres männlichen Publikums ist farbfahlsichtig und kann mit der Anweisung Zum Bestellen drücken Sie bitte auf den grünen Knopf nichts anfangen.	
Tag 3: Sauberer Code	11
Sauberer Code ist nicht nur der Zeitvertreib von HTML-Puristen, sondern bringt schon von alleine ein Mindestmaß an Barrierefreiheit mit sich.	
Tag 4: Sprachliche Besonderheiten	19
Die Kennzeichnung von Besonderheiten wie Sprachwechsel und Abkürzungen ist eine der wenigen reinen Accessibility-Vorgaben der BITV.	
Tag 5: Tabellen	23
Layout-Tabellen haben mittlerweile ihr Verfallsdatum deutlich überschritten.	
Tag 6: Fallback-Lösungen	27
Ein Blick in die Statistik Ihrer Website enthüllt die Vielzahl der unterschiedlichsten Browser aus verschiedenen Generationen, die auf Ihre Website zugreifen.	
Tag 7: Dynamik	31
Das Web ist weit mehr als nur ein Sammlung wohlstrukturierter statischer Texte. Interaktive Elemente sind oft der bessere Weg, um manche Nutzergruppen zu erreichen.	
Tag 8: Zugang	35
In dieser Folge geht es um die Zugänglichkeit der Inhalte, die über das bloße HTML oder Bilder hinausgehen und eigene Schnittstellen zur Ausgabe und Bedienung haben.	
Tag 9: Unabhängigkeit	38
Wenn Ihre Seiten für Sprachausgaben, Braillezeilen und andere assistive Technologien zugänglich sind, werden sie auch für Kiosksystemen ohne Tastatur oder mobile Endgeräten bedienbar sein.	
Tag 10: Kompatibilität mit assistiven Technologien	42
Nicht jedes Ausgabegerät verfügt über die selben Möglichkeiten wie der Rechner, auf dem die Website entwickelt wurde.	
Tag 11: Standards	47
Die Verwendung von offenen Standards garantiert nicht nur die Interoperabilität Ihres Internetangebots, sondern spart bei Wartung und Pflege bares Geld.	
Tag 12: Orientierung	50
Vielen Benutzern assistiver Technologien wird der Zugang bereits durch die mangelnde Übersichtlichkeit einer Website verwehrt.	
Tag 13: Gebrauchstauglichkeit	53
Klassische Usability-Faktoren haben ebenso einen starken Einfluss auf die Barrierefreiheit eines Angebotes.	
Tag 14: Verständnis	57
Selbst wenn die Prüfprogramme auf keine groben Schnitzer hinweisen, ist eine Website damit noch lange nicht für alle Besucher tatsächlich auch benutzbar.	
Wie geht es weiter?	60

Tag 1: Alternativen

Am ersten Tag unserer BITV-Serie geht es um einige grundlegende Dinge, die schon immer zum "Guten Ton" einer Website gehörten. Viele Bedingungen der BITV erfüllt man schon fast automatisch, sobald man sauberen Code nach den etablierten Standards verwendet. Andere Bedingungen sind da schon aufwändiger in der Umsetzung – mehr dazu am Ende. Zunächst aber die wohl am einfachsten zu erfüllende Vorgabe – Alternative Texte:

Anforderung 1		Für jeden Audio- oder visuellen Inhalt sind geeignete äquivalente Inhalte bereitzustellen, die den gleichen Zweck oder die gleiche Funktion wie der originäre Inhalt erfüllen.
Bedingung 1.1	Prio. 1	Für jedes Nicht-Text-Element ist ein äquivalenter Text bereitzustellen. Dies gilt insbesondere für: Bilder, graphisch dargestellten Text einschließlich Symbolen, Regionen von Imagemaps, Animationen (z. B. animierte GIFs), Applets und programmierte Objekte, Zeichnungen, die auf der Verwendung von Zeichen und Symbolen des ASCII-Codes basieren (ASCII-Zeichnungen), Frames, Scripts, Bilder, die als Punkte in Listen verwendet werden, Platzhalter-Graphiken, graphische Buttons, Töne (abgespielt mit oder ohne Einwirkung des Benutzers), Audio-Dateien, die für sich allein stehen, Tonspuren von Videos und Videos.

Was heißt das?

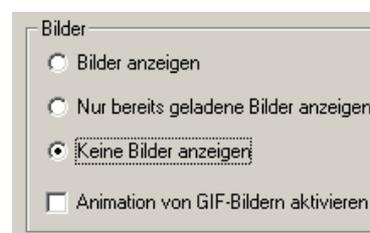
In dieser Anforderung geht es primär darum, das Vorhandensein und die Qualität der seit HTML 4 verbindlich vorgeschriebenen alternativen Texte zu Bildern, Imagemaps und anderen Nicht-Text-Elementen wie `<object>` zu überprüfen. Eigentlich ein einfaches Unterfangen, da die meisten Editoren hierfür separate Eingabefelder beim Einfügen von Objekten vorgesehen haben. Aber spätestens bei der Frage, was denn nun ein äquivalenter Text sei, scheiden sich die Geister.

Die Vorgabe beschränkt sich aber nicht nur auf alternative Texte zu grafischen Elementen. Wenn man den Text genau liest, stellt man fest, das hier alternative Texte zu sämtlichen optionalen Formaten, die nicht im Text direkt beschrieben werden, gefordert werden. Streng genommen muss also **der gesamte grafische Inhalt** einer Webseite zusätzlich nochmals in Textform hinterlegt oder beschrieben werden. Der Einfachheit halber haben wir die folgenden Tests in Bilder und deren Alternativen sowie sonstige optionale Formate aufgeteilt.

Wie können Sie das testen?

Bei Bildern und anderen grafischen Elementen können Sie diese Anforderung ganz leicht überprüfen, in dem Sie in den Voreinstellungen Ihres Browsers (bei Opera ist dies z.B. F12) das Laden von Bildern komplett unterbinden.

Noch einfacher geht es, wenn Sie sich in Opera den Button Bilder zeigen in der Toolbar installieren. Die Einstellungen für die Toolbar finden Sie im Menü View unter dem Punkt Customize Toolbars. Danach können Sie per Knopfdruck zwischen den Ansichten mit und ohne Bildern hin- und herschalten.



Der Browser zeigt dann an Stelle des Bildes den alternativen Text an, sofern einer hinterlegt ist. Fehlt das `alt`-Attribut komplett, wird lediglich `[Image]` angezeigt. Hier wird deutlich, warum man dekorative

Elemente oder sog. Spacer, die selbst keine Inhalte transportieren, mit einem leeren `alt=""` verstecken sollte. In diesem Fall versucht der Browser das Bild (bzw. den Platz, den es einnehmen würde) nämlich erst gar nicht zu rendern, so dass der Lesefluss nicht gestört wird.

Wenn sich Ihnen dann beim Lesen ohne Bilder noch **der komplette Inhalt und alle Funktionen** der Seite erschließen, können Sie diesen Punkt abhaken. Sie sehen nämlich genau die Inhalte, die auch ein Screenreader oder Mobile Devices vorlesen würde.

Ein Beispiel:

Ein einfach nachzuvollziehendes Beispiel hierfür ist das Logo der Aktion Mensch auf diesen Seiten, das im sogenannten "Masthead" erscheint. In grafischen Browsern erscheint unten links auf jeder Seite der Hinweis "Einfach für Alle" ist eine Initiative der" gefolgt von einer Grafik mit dem Logo der Aktion Mensch.



Üblicherweise würde der alternative Text für die Grafik nun heißen: "Logo der Aktion Mensch" – allerdings würde dann ein Screenreader auch genau das vorlesen: "Einfach für Alle ist eine Initiative der Logo der Aktion Mensch".



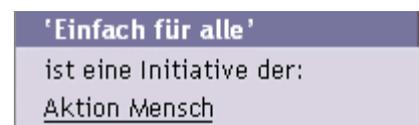
Der folgende Screenshot zeigt nun die Darstellung des an Stelle dessen verwendeten alternativen Textes, der den tatsächlichen Inhalt des Logos, nämlich den Text "Aktion Mensch" wiedergibt. Hier wurde zur Kontrolle der Qualität der `alt`-Texte das Laden der Bilder im Browser unterbunden, das Ergebnis ist immer noch lesbar und diesmal auch sinnvoll.



In halbwegs modernen Browsern können Sie sogar noch den Zeichensatz für die alternativen Texte per CSS bestimmen, so dass diese nicht in der voreingestellten Standardschrift (also wie in der vorhergehenden Abbildung in Times) erscheinen. Die einfache Anweisung

```
img{font-family:"Trebuchet MS",sans-serif}
```

bringt den Browser auf den richtigen Kurs, so dass auch Besucher, die keine Bilder laden, in den vollen Genuss Ihrer Gestaltung kommen können:



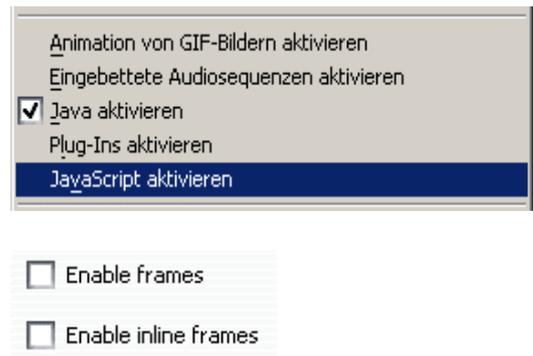
Bei anderen optionalen Formaten ist der Test etwas aufwändiger, aber auch mit den Bordmitteln aktueller Browser zu schaffen. Nochmal zurück zur Anforderung:

"[...] Animationen (z.B. animierte GIFs), Applets und programmierte Objekte, [...] Frames, Scripts, [...] Platzhalter-Graphiken, graphische Buttons, Töne (abgespielt mit oder ohne Einwirkung des Benutzers), Audio-Dateien, die für sich allein stehen, Tonspuren von Videos und Videos."

Hier helfen wieder die Schnelleinstellungen von Opera (F12). Deaktivieren Sie einfach alle dort aufgelisteten Formate und laden dann Ihre Website.

Ausnahme: in den Schnelleinstellungen ist leider keine Funktion, um Frames zu deaktivieren. Dieser versteckt sich in den allgemeinen Einstellungen des Programms unter Datei / Einstellungen / Seitenlayout.

Problematisch wird das Testen der Vorgabe nur, wenn es um synchronisierte Untertitel (closed captioning) von Videosequenzen geht. Da diese meist im Video selbst enthalten sind, müssen Sie diese natürlich mit aktivierten Plug-Ins kontrollieren.



Bedingung 1.2	Prio. 1	Für jede aktive Region einer serverseitigen Imagemap sind redundante Texthyperlinks bereitzustellen
---------------	---------	---

Was heißt das?

Serverseitige Imagemaps waren kurzfristig gegen Mitte der neunziger Jahre des letzten Jahrhunderts in Mode, als manche Browser (Netscape 1.0) die gerade erfundenen Client-seitigen Imagemaps noch nicht verstanden ([s. hierzu Bedingung 1.5](#)), daher können sie diesen Punkt getrost überspringen.

Bedingung 1.5	Prio. 2	Für jede aktive Region einer clientseitigen Imagemap sind redundante Texthyperlinks bereitzustellen.
Bedingung 9.1	Prio. 1	Es sind clientseitige Imagemaps bereitzustellen, es sei denn die Regionen können mit den verfügbaren geometrischen Formen nicht definiert werden.

Was heißt das?

Für den Fall, dass ein Browser keine Bilder anzeigen kann oder will, sieht der HTML-Standard ein `alt`-Attribut für jede aktive Region einer Imagemap vor. Diese Attribute sind vergleichbar mit den alternativen Texten herkömmlicher Bilder, nur ist die Unterstützung für diese speziellen alternativen Texte in den verschiedenen Browsern so inkonsistent, dass die redundanten Texthyperlinks einfach seit jeher zum guten Ton gehören. Zudem haben diese Texthyperlinks den Vorteil, im Gegensatz zu Imagemaps wirklich mit jedem User Agent bedienbar zu sein.

Wie können Sie das testen?

Dies geht natürlich nur mit einer visuellen Überprüfung, die Sie aber mit jedem beliebigen Browser vornehmen können. Sind sie vorhanden und auch noch sinnvoll getextet, haben Sie diesen Test bestanden.

Ein Beispiel:

Bei einer Weltkarte, auf der die Kontinente anklickbar sind, sollte in unmittelbarer Nähe der Grafik noch zusätzliche Texthyperlinks ([Antarktis](#) | [Australien](#) | ... etc.) bereitgestellt werden.

Achten Sie bei diesen redundanten Links allerdings darauf, dass der genaue Text, der verlinkt wird, nicht mehrfach als Link auf einer Seite enthalten ist. Nicht nur Prüfprogramme wie Bobby meckern dann, auch Screenreader, die Links zusammenfassen und getrennt darstellen können, werden durch eine Vielzahl gleichlautender Links verwirrt.

Bedingung 1.3	Prio. 1	Für Multimedia-Präsentationen ist eine Audio-Beschreibung der wichtigen Informationen der Videospur bereitzustellen.
Bedingung 1.4	Prio. 1	Für jede zeitgesteuerte Multimedia-Präsentation (insbesondere Film oder Animation) sind äquivalente Alternativen (z.B. Untertitel oder Audiobeschreibungen der Videospur) mit der Präsentation zu synchronisieren.

Eine der wohl schwierigsten Bedingungen aus der gesamten BITV. Wenn man die anderen Bedingungen der BITV auf diese hier anwendet, würde das bedeuten, dass es von dem Audiotrack dann auch eine Textmitschrift geben müsste. Mal abgesehen von den technischen Schwierigkeiten, wenn der Screenreader dann die Textmitschrift einer bereits laufenden Audiodeskription noch zusätzlich parallel vorliest – müssten dann nicht gleich auch noch ehrlicherweise alle Inhalte gebärdet oder zumindest in DGS-taugliches Format übersetzt werden?

Hier ist sicher noch Klärungsbedarf vorhanden. Im Rahmen der Aktion Grundgesetz und Einfach für Alle wird die Aktion Mensch in der nächsten Zeit [Inhalte in Deutsche Gebärdensprache](#) übersetzen und die gefundenen Lösungen zur Diskussion stellen. Eine erste Fassung von Inhalten in DGS finden Sie in unserem Presseportal bei [Fakten zur Aktion Mensch](#).

Tag 2: Farben

Wenn Sie die Tests des heutigen Tages an Ihren eigenen Seiten nachvollziehen wollen, empfehlen wir Ihnen noch die Lektüre der Einleitung zu dieser Serie. Dort finden Sie am Ende des Artikels eine [Liste mit den benötigten Tools](#) und Adressen, wo Sie diese herunterladen können. Nun aber zur heutigen Anforderung aus der BITV:

Anforderung 2		Texte und Graphiken müssen auch dann verständlich sein, wenn sie ohne Farbe betrachtet werden.
Bedingung 2.1	Prio. 1	Alle mit Farbe dargestellten Informationen müssen auch ohne Farbe verfügbar sein, z.B. durch den Kontext oder die hierfür vorgesehenen Elemente der verwendeten Markup-Sprache.

Was heißt das?

Circa zehn Prozent Ihres männlichen Publikums sind farbfeldsichtig, das heißt diese Besucher verfügen über eine ungenügende Differenzierung von bestimmten Farben. Welche Farben dies sind, hängt von der Art der Farbfeldsichtigkeit ab, da es mehrere, höchst unterschiedliche Varianten gibt. Die wohl am häufigsten anzutreffende Fehlsichtigkeit ist der Rot-Grün-Verwechslung; diese Besucher können mit der Anweisung „Zum Bestellen drücken Sie bitte auf den grünen Knopf“ unter Umständen nichts anfangen.

Wie können Sie das testen?

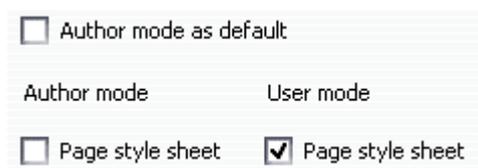
Falls Sie [Mozilla](#) benutzen, sollten Sie sich die [Preferences Toolbar von xulplanet.com](#) installieren. Mit dieser Toolbar bekommen Sie eine Vielzahl von Einstellungsmöglichkeiten in Ihr Browser-Interface, die nicht nur das Testen, sondern auch die Entwicklung moderner Webseiten stark erleichtern.



In dieser konfigurierbaren Toolbar finden Sie unter anderem Checkboxes, um Ihre Website komplett ohne Farben, ohne Fontangaben oder ohne Bilder zu betrachten. Sie können auch JavaScript ausschalten, um Ihre Navigation oder Formulare zu testen oder das Browserfenster auf bestimmte Größen skalieren und so kleinere Monitore simulieren.

Der eigentliche Test ist dann ganz simpel: schalten Sie in der Toolbar das Laden von Farben über die Checkbox Colors aus und laden Sie Ihre Website. Wenn sie nun noch an alle Inhalte kommen und alle Funktionen ausführen können, haben Sie diesen Test bestanden.

Alternativ können Sie mit Opera auch auf jegliche Farbangaben verzichten, indem sie in den Voreinstellungen alle entsprechenden Anweisungen unterbinden. Wenn Sie alle Checkboxes unter Benutzermodus aktivieren und alle Checkboxes unter Autorenmodus deaktivieren, bekommen Sie in Opera allerdings im Gegensatz zu der oben für Mozilla beschriebenen Methode eine nackte HTML-Seite ohne jegliche Formatierungen.



Wenn sie bei Ihren Tests öfters zwischen Autoren- und Benutzermodus hin- und herschalten müssen, sollten Sie sich die entsprechende Funktion der Einfachheit halber gleich über "Symbolleisten anpassen" in der Toolbar installieren.



Ein Beispiel:

Nehmen wir wieder die Seiten von 'Einfach für alle'. Hier finden Sie in grafischen Browsern auf der rechten Seite die Navigationsleiste, in der die vorhandenen Unterbereiche durch einen farbigen Balken voneinander getrennt sind. In diesen Balken steht in negativem Text die jeweilige Überschrift des Unterbereichs.

Verhindert man nun die Farbangaben, so erscheint der gesamte Text in schwarz auf weiß, die Links sind nun blau auf weiß. Die Struktur ist jedoch trotzdem erkennbar, da für die Überschriften der Bereiche die entsprechenden Tags aus HTML (in diesem Fall <h4>) verwendet wurden. Der Test ist also, zumindest was die Navigation angeht, bestanden.



Bedingung 2.2	Prio. 1	Bilder sind so zu gestalten, dass die Kombinationen aus Vordergrund- und Hintergrundfarbe auf einem Schwarz-Weiß-Bildschirm und bei der Betrachtung durch Menschen mit Farbfehlsichtigkeiten ausreichend kontrastieren.
Bedingung 2.3	Prio. 2	Texte sind so zu gestalten, dass die Kombinationen aus Vordergrund- und Hintergrundfarbe auf einem Schwarz-Weiß-Bildschirm und bei der Betrachtung durch Menschen mit Farbfehlsichtigkeiten ausreichend kontrastieren.

Diese Anforderungen sind mit den gängigen Mitteln der heutigen Betriebssysteme und Browser nicht zu testen, daher müssen wir hier auf Online-Tools oder Grafikprogramme ausweichen. Die einfachste Methode ist hier, einen Screenshot Ihrer Website in einem Grafikprogramm wie Photoshop oder Fireworks zu entfarben. In Fireworks geht dies z.B. über den Befehl Graustufenkonvertierung, in Photoshop finden Sie den Befehl je nach Version unter Bild modifizieren / Graustufen.

Etwas genauer als die reine Konvertierung nach Graustufen ist die tatsächliche Simulation von Farbfehlsichtigkeiten, wie sie bei www.vischeck.com möglich ist. Dieses kostenlose Online-Tool erstellt eine Grafikdatei Ihrer Website mit den Farben, wie sie Besucher mit den gängigsten Farbfehlsichtigkeiten sehen würde.



Noch genauer ist die Konvertierung, die unter colorfilter.wickline.org angeboten wird. Nach Eingabe der URL Ihrer Seite bekommen Sie keine Grafikdateien, sondern umgefärbte HTML-Seiten zurück und haben die Wahl zwischen einer Vielzahl unterschiedlichster Farbfehlsichtigkeiten.



Tipp: Beachten Sie, dass alle diese Tests, insbesondere die visuellen Überprüfungen in einer Position etwa eine Armlänge vom Monitor entfernt durchgeführt werden sollten (Normalsichtigkeit vorausgesetzt). Wenn Sie Inhalte nur noch erkennen können, indem Sie sich näher zum Monitor hinbewegen, werden dies auch viele Ihrer Benutzer machen müssen. Das z.B. bedeuten, dass Ihre Farben zu wenig kontrastreich sind, oder die Inhalte zu filigran, um in normaler Sitzposition noch wahrgenommen zu werden.

Angesichts der Häufigkeit der Farbfehlsichtigkeiten sind die Chancen gut, dass sich in Ihrem Bekanntenkreis ein Betroffener befindet. Bitten Sie diesen doch einfach um einen kurzen Test Ihrer Seiten. Idealerweise geschehen [solche Tests](#) bereits in der Designphase einer Website, also zu dem Zeitpunkt, wenn das Farbschema mit dem Kunden abgestimmt wird. Mit CSS lassen sich Farbänderungen zwar in Sekundenschnelle durchführen, nur wird dadurch der Abstimmungsprozess mit dem Kunden natürlich nicht beschleunigt. Und nichts ist schlimmer, als wenn Fehler erst nach dem Launch vom Kunden und dessen Besuchern gefunden werden. Also: frühzeitig testen!

Tag 3: Sauberer Code

Sauberer Code ist nicht nur der Zeitvertreib von HTML–Puristen, sondern bringt schon von alleine ein Mindestmaß an Barrierefreiheit mit sich. Die Validierung der Dokumente schafft die solide Basis für jedes Webangebot. Durch die saubere Trennung von Gestaltung und Inhalt verringert sich nicht nur die Ladezeit beim Benutzer, Ihre Seiten werden auch offen für alle möglichen Ausgabeformen.

Die Zeiten, wo [proprietäres Markup](#) nötig war, um eine Webseite überhaupt lauffähig zu machen, sind längst vorbei. Denn auch die [Browserhersteller](#) haben den Trend längst erkannt und bringen seit Jahren nur noch Produkte auf den Markt, die dem Ideal der vollständigen Unterstützung der Standards immer näher kommen.

Anforderung 3	Markup-Sprachen (insbesondere HTML) und Stylesheets sind entsprechend ihrer Spezifikationen und formalen Definitionen zu verwenden.	
Bedingung 3.1	Prio. 1	Soweit eine angemessene Markup-Sprache existiert, ist diese anstelle von Bildern zu verwenden, um Informationen darzustellen.

Was heißt das?

Letters are things, not pictures of things

Der berühmte Typograf *Eric Gill* hatte 1940 sicher anderes im Sinn, als er dieses Zitat von sich gab, aber es trifft im Kern die heutigen Gegebenheiten im Web: Zu viele Texte werden in Bildern versteckt.

Wie können Sie das testen?

Hier hilft wieder das Gelernte von [Tag 1 dieses Tutorials](#): Schalten Sie sämtliche optionalen Inhalte einer HTML–Seite ab und überprüfen Sie, ob der Sinn und Zweck der Seite noch transportiert wird und der vollständige Inhalt immer noch vorhanden und erreichbar ist. Dies betrifft nicht nur Zusatzformate wie Java, Flash oder JavaScript, auch in Bildern können sich wesentliche Inhalte verstecken, die sich **ohne** diese Bilder nicht mehr erschließen lassen.

Ein Beispiel:

Im Kopf der EfA–Seiten finden Sie eine Grafik mit dem Inhalt "Informationen zum barrierefreien Webdesign". Wenn sie nun Style Sheets deaktivieren, finden Sie an der gleichen Stelle nun einen Absatz mit dem identischen Textinhalt der Grafik. Der Trick hierbei ist, dass für grafische Browser an dieser Stelle per CSS ein Bild eingefügt wurde; für Textbrowser ohne Grafikfähigkeiten oder Sprachausgaben ist der Bildinhalt nochmals als Text hinterlegt.

Bedingung 3.2	Prio. 1	Mittels Markup-Sprachen geschaffene Dokumente sind so zu erstellen und zu deklarieren, dass sie gegen veröffentlichte formale Grammatiken validieren.
---------------	---------	---

Was heißt das?

Eine kurze Anforderung, die es aber in sich hat. Dies bedeutet nichts anderes, als dass Ihre Seiten gültig, sauberen Code nach den [HTML–Standards des W3C](#) aufweisen müssen. *Ohne wenn und aber*.

Das beginnt übrigens bereits mit der sogenannten Document Type Definition (DTD), nach der sich ein HTML-Dokument als solches outet – nichts anderes ist mit und zu deklarieren gemeint. Es reicht also nicht, HTML-Dokumente mit dem Suffix .htm oder .html zu speichern und den Rest dem Browser zu überlassen. Auch im Dokument selbst steht als Erstes, nach welcher der verschiedenen (X)HTML-Geschmacksrichtungen sich Ihr Dokument richtet.

Dies hilft bei der Fehlersuche, wenn sie Ihre Seiten durch eines der verschiedenen Prüfprogramme wie z.B. validator.w3.org kontrollieren lassen. Dies bringt aber moderne Browser auch dazu, sich an die Regeln zu halten und die Angaben z.B. im CSS genau so umzusetzen, wie vom Seitenbauer gewünscht. Verlassen Sie sich aber nicht auf die DTD's, die von manchen Autorentools eingesetzt werden – diese sind oftmals fehlerhaft. Eine [Liste der korrekten DTD's](#) finden Sie beim W3C.

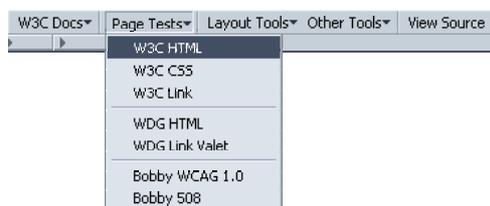
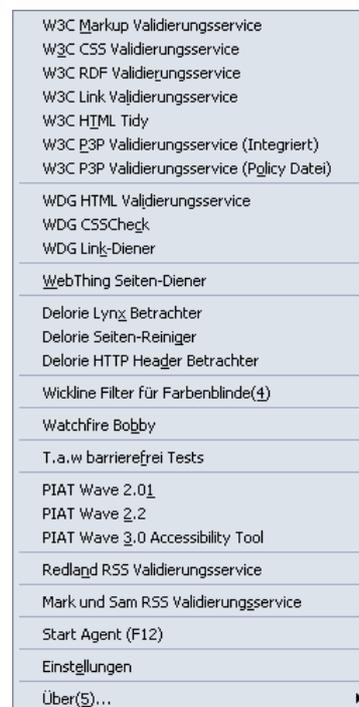
Damit aber nicht genug, auch der Rest des HTML-Codes muss bestimmten Qualitätskriterien genügen. Wie Sie diese überprüfen können, erfahren Sie im nächsten Abschnitt.

Wie können Sie das testen?

Ein sehr sinnvolles Werkzeug für die heute zu bewältigenden Tests ist [Checky für Mozilla](#) und seine Verwandten. Dieses Werkzeug übergibt nach der Installation Ihre Seiten über ein Kontextmenü an eine Vielzahl von Online-Tools. Einfach mit der rechten Maustaste bzw. Ctrl. auf eine Seite klicken und im Pop-Up Menü unter "Checky" das gewünschte Tool auswählen. Die Ergebnisse werden alle in einem neuen Tab geöffnet, so dass Sie diese nacheinander lesen oder speichern können.

Die Seiten müssen dafür natürlich erreichbar auf einem Server im Netz liegen, mit lokalen Dateien kann Checky nichts anfangen. Zudem haben die meisten Prüftools Probleme, an Dateien in Passwort-geschützten Bereichen zu kommen. Daher sollten Sie die Seiten zumindest für die Zeit der Tests auf einen öffentlich erreichbaren Server installieren.

In der gleichen Liga spielt ein anders Tool für Entwickler: die PNH Developer Toolbar von Chris Casciano, die Sie von placenamehere.com/pnhtoolbar aus in Ihren Mozilla installieren können. Hier liegt der Focus allerdings weniger auf nachträglichen Tests (wenn das Kind meistens schon in den Brunnen gefallen ist), sondern die Hilfen setzen bereits bei der Entwicklung an.



Alternativ können Sie natürlich auch auf lokale Syntax-Checker zurückgreifen, die in so gut wie allen Autorentools wie Dreamweaver oder GoLive bereits eingebaut sind. Andere Editoren wie [TopStyle](#), die näher am Code arbeiten, bieten oftmals die Möglichkeit, den Code der erstellten Seiten an eines der Online-Prüftools zu übergeben. Links zu diesen Prüftools finden Sie übrigens in Hülle und Fülle in der EfA-Linkmaschine im Bereich Prüfprogramme.

Ein Beispiel:

Dieser Link übergibt die aktuelle Seite an den HTML-Validator des W3C:



Dieser Link übergibt das Style Sheet der aktuellen Seite an den CSS-Validator des W3C:



Bedingung 3.3	Prio. 1	Es sind Stylesheets zu verwenden, um die Text- und Bildgestaltung sowie die Präsentation von mittels Markup-Sprachen geschaffener Dokumente zu beeinflussen.
---------------	---------	--

Was heißt das?

Nichts anderes als die saubere Trennung von Inhalt und Verpackung. Das HTML Ihrer Seiten sollte lediglich die Inhalte beinhalten und deren Struktur festlegen, die Gestaltung für alle denkbaren Ausgabeformen wird hingegen mit Cascading Style Sheets (CSS) erreicht.

Konkret bedeutet das, dass Sie z.B. Ihre Überschriften im HTML-Code als solche auszeichnen (also mit h1-h6) und die Gestaltung dann nicht auch noch im HTML regeln, sondern über eine (vorzugsweise externe) CSS-Datei.

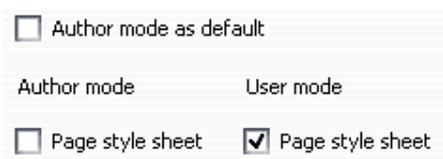
Wie können Sie das testen?

ToggleCSS von Tantek Celik ist ein wunderbar simples Bookmarklet, das sie einfach in die Favoritenleiste Ihres Browsers legen können. Der Vorteil dieser Bookmarklets (oder Favelets) besteht darin, dass es simple JavaScripts sind, die bestimmte Aktionen mit Ihren Seiten ausführen. Im konkreten Fall werden durch einen Klick sämtliche Formatierungen, die aus einem Style Sheet kommen, aus der Seite entfernt. Was dann noch übrig bleibt, ist im Idealfall die pure Struktur, wie sie im HTML steht. Weitere fleißige Helferlein dieser Art finden Sie im Netz beim W3C oder bei www.squarefree.com/bookmarklets.

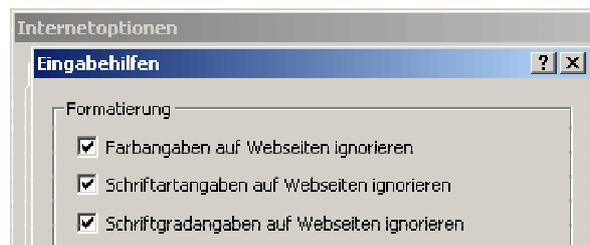


Sie können in Ihrem Browser natürlich auch das Laden von CSS komplett unterbinden. Sinnvoll ist dies um z.B. einen ganzen Satz Dateien nacheinander zu testen, ohne jedes mal die bereits geladenen Styles wieder händisch zu entfernen. In Opera finden Sie die entsprechenden Anweisung unter Datei / Einstellungen / Seitenlayout / Autorenmodus / Stil der Seite.

Wenn Sie hier alle Häkchen unter "Stil der Seite" entfernen, sehen Sie, wie viel (oder wie wenig) Formatierung aus der HTML-Seite selbst kommt. Erscheint Ihre Seite nun nicht in der Standardschrift des Webbrowsers (also höchstwahrscheinlich Times) oder sind immer noch andere Vorder- und Hintergrundfarben als schwarz und weiß zu sehen (blaue Links natürlich ausgenommen), so verstecken sich noch Formatierungsanweisungen im HTML-Code.



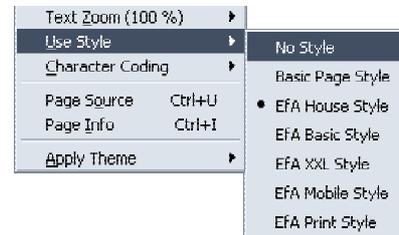
Das Pendant dazu im Internet Explorer finden Sie unter Extras / Internetoptionen / Eingabehilfen. Aller-



dings wird hier durch die Vorgabe "ignorieren" nicht die gesamte CSS-Gestaltung entfernt, sondern nur die Angaben zu Schrifttypen und -größen sowie die Farbangaben. Die Positionierung der einzelnen Seitenelemente per CSS bleibt nach wie vor erhalten.

Ein Beispiel:

Falls Sie diese Seiten hier gerade in einem aktuellen Browser wie Mozilla lesen, schalten Sie doch einfach mal alle Style Sheets ab. Im Menü View / Use Style ist dies der Eintrag "No Style". Schon erhalten Sie den reinen ungestalteten Text, der aber nach wie vor über seine eingebaute Struktur verwertbar ist.



Bedingung 3.4	Prio. 1	Es sind relative anstelle von absoluten Einheiten in den Attributwerten der verwendeten Markup-Sprache und den Stylesheet-Property-Werten zu verwenden.
---------------	---------	---

Was heißt das?

Sämtlichen Inhalten einer HTML-Seite können nicht nur Informationen über ihr Aussehen, sondern auch über ihre Größe und Position mitgegeben werden. Der Einfachheit halber gehen wir mal davon aus, dass dies bei Ihren Seiten bereits über CSS und nicht z.B. über das `font`-Tag geschieht. Ausnahme: Tabellen – hier können Sie natürlich weiterhin Bemaßungen im HTML-Code angeben und trotzdem die BITV erfüllen. Sie sollten nur darauf achten, dass diese Bemaßungen nicht in absoluten Einheiten (also Pixel, Punkt, cm, mm oder Zoll), sondern in relativen Einheiten (also em, ex oder Prozent) gesetzt werden. Nur so passen sich Schriftgrößen an die Voreinstellungen des Benutzers an oder Tabellen an die Größe des Browserfensters.

Ausnahmen sollte es natürlich auch hier geben: Größenangaben für Gestaltungselemente wie Ränder können Sie natürlich weiterhin für die Monitordarstellung in Pixel angeben, auch eingebettete Bilder haben generell Pixeldimensionen.

Noch eine Ausnahme: In einem [Style Sheet für die Druckausgabe](#) einer Seite ist es natürlich sinnvoller die Angaben für Dimensionierung und Platzierung in cm oder mm anzugeben und die Angaben für Schriftgrade in typografischen Punkten (pt). Der Zielrechner muss sämtliche Werte beim Ausdruck ohnehin in diese Maßeinheiten umrechnen.

Wie können Sie das testen?

Zur Überprüfung brauchen Sie einen Internet Explorer (egal welche Version) unter Windows (egal welche Version). Dieser Browser ist in der Standardeinstellung als einziger nicht in der Lage, Texte, deren Größe in Pixel definiert wurde, zu skalieren. Wenn in Ihren Seiten also z.B. der Fließtext in 9 Pixel großer Arial definiert wurde, so können Sie diese Größe im IE nicht mehr verstellen. Am einfachsten lässt sich dies überprüfen, indem Sie im Menü Ansicht / Symbolleiste / Anpassen die Schaltfläche "Größe" hinzufügen. Stellen Sie einfach vom voreingestellten Wert auf "Sehr groß" oder "Sehr klein". Wenn sich nun die **gesamten** Schriftgrößen an Ihrer Seite ändern, haben Sie diesen Test bestanden.





Ein Beispiel:

Ein praktisches Anwendungsbeispiel ist der auf diesen Seiten verwendete [EfA-fontsizer](#) am Beginn der Navigation. Da alle Bemaßungen auf dieser Website relativ zu den Gegebenheiten des Zielbrowsers gesetzt sind, wird das Layout auch bei einer Änderung der Werte durch den Benutzer nicht zerstört und der Gesamteindruck bleibt erhalten.

Bedingung 3.5	Prio. 1	Zur Darstellung der Struktur von mittels Markup-Sprachen geschaffener Dokumente sind Überschriften-Elemente zu verwenden.
Bedingung 13.8	Prio. 2	Es sind aussagekräftige Informationen am Anfang von inhaltlich zusammenhängenden Informationsblöcken (z.B. Absätzen, Listen) bereitzustellen, die eine Differenzierung ermöglichen.

Was heißt das?

Sie kennen das aus Ihrer Textverarbeitung: Man kann Überschriften entweder als großen fetten Text anlegen, oder man kann eine Formatvorlage benutzen, um eine Überschrift als Überschrift 1, Überschrift 2 oder Überschrift 3 zu definieren. Beides sieht auf dem Monitor und im Ausdruck gleich aus, nur bedeutet der erste Weg wesentlich mehr Arbeit. Zudem kann Ihre Textverarbeitung die Überschriften in der Dokumentstruktur als solche ausgeben und ihnen damit die Navigation im Dokument vereinfachen.

Nichts anderes geschieht im Web, wenn sie Ihre Überschriften mit den dafür vorgesehenen Tags <h1–<h6 auszeichnen. Nicht nur [Screenreader wie JAWS](#) können nun die Struktur einer Seite besser wiedergeben und es dem Benutzer ermöglichen, sich von Überschrift zu Überschrift zu bewegen. Auch immer mehr grafische Browser können Seiten auf diese Art zusammenfassen.

Wie können Sie das testen?

Hier kommt wieder die oben erwähnten Eingabehilfen im Internet Explorer zum Einsatz. Schalten Sie einfach alle Angaben zu Schriftgrad und –art ab und lesen Sie Ihren Text. Wenn die Überschriften nun genau so aussehen wie Ihr Fließtext, bedeutet dies, das diese nicht korrekt ausgezeichnet sind. Denn auch ohne Angaben aus dem HTML oder CSS werden Überschriften in grafischen Browsern entsprechend Ihrer Bedeutung dargestellt, d.h. h1 sehr groß und fett, h4 oder h5 dagegen in der Größe von normalem Fließtext.

Alternativ können Sie auch einen der Browser zu Rate ziehen, die bereits die Navigation per Überschriften beherrschen. Dies ist nicht nur wie üblich der Mozilla, auch andere Browser wie Opera, iCab usw. erleichtern ihren Nutzern so die Navigation.



Für diese Tests (und weitere an den kommenden Tagen) sollten Sie sich die [Context Menu Extension](#) für Mozilla installieren. Mit dieser Erweiterung bekommen Sie einen neuen Menüeintrag, hinter dem sich viele komfortable Navigationshilfen verbergen.

Laden Sie Ihre Seite und versuchen Sie, nur über das Springen von Überschrift zu Überschrift zu navigieren. Dieses Verhalten entspricht den Möglichkeiten von assistiven Technologien wie JAWS oder HPR, Überschriften aus dem Kontext losgelöst darzustellen. Wenn nun die Überschriften in der richtigen Reihenfolge (also nicht zuerst irgendwelche Unterüberschriften und dann mittendrin die wichtigste Überschrift) kommen, so haben Sie diesen Test bestanden.

Ein Beispiel:

Bevor CSS starke Verbreitung fand, wurden Überschriften üblicherweise so ausgezeichnet:

```
<br><br>
<font size="7"><b>
<font face="Helvetica">
<font color="#ff0000">
Gro&szlig;e fette &Uuml;berschrift
</font>
</b></font>
</font>
<br><br>
```

Mit CSS und HTML 4 sollte die wichtigste Überschrift Ihrer Seite nun so aussehen.

```
<h1Gro&szlig;e fette &Uuml;berschrift</h1
```

Die Formatierung über CSS bringt nun das gewünschte Aussehen:

```
h1 {
  font : bold 2em/150% Helvetica, sans-serif;
  color: #f00;
}
```

So ziemlich alle Browser bieten die Möglichkeit, den Quelltext einer Seite anzuzeigen. Suchen Sie im Code einfach nach dem Text einer Überschrift. Wenn diese nun so aussieht wie in unserem zweiten Beispiel (h1, h2, . . .), haben Sie den Test bestanden.

Bedingung 3.6	Prio. 1	Zur Darstellung von Listen und Listenelementen sind die hierfür vorgesehenen Elemente der verwendeten Markup-Sprache zu verwenden.
---------------	---------	--

Was heißt das?

Wie bei den meisten HTML-Elementen lassen sich auch Textauszeichnungen zu Zwecken missbrauchen, für die sie nicht gedacht waren. So kann man Listen und Aufzählungen auch durch Einrückungen, Zeilenschaltungen und eingebettete Bilder (Bullits) erreichen. Diesen fehlt aber dann natürlich die eingebaute Logik, die richtige Listen mit den Tags ``, ``, oder `<dl>` bereits mit sich bringen. Ein Beispiel für letzteren Tag ist das [Glossar zur BITV](#).

Wie können Sie das testen?

Dieser Test verläuft so ähnlich wie der vorhergehende Test mit den Überschriften: Schalten Sie alle Formatierungen in Ihrem Browser ab und kontrollieren Sie dann, ob eventuell vorhandene Auflistungen noch eingerückt sind und mit einem Listensymbol (quadratisches oder rundes Bullit) oder einer Zahl versehen sind. Ist dies so, dann sind es echte Listen und Sie haben diesen Test bestanden.

Ein Beispiel:

Ein anderer, durchaus zulässiger Anwendungszweck für Listen sind sogenannte Codelistings, wie sie bei EfA in den eher technisch orientierten Artikeln eingesetzt werden ([Beispiel-Listing](#)). In diesem Fall sind es nummerierte Listen (``), die sich so verhalten, wie es Quelltext in einem Editor machen würde.

Bedingung 3.7	Prio. 1	Zitate sind mittels der hierfür vorgesehenen Elemente der verwendeten Markup-Sprache zu kennzeichnen.
---------------	---------	---

Was heißt das?

HTML kennt für Zitate und Quellenangaben verschiedene Tags, die hierfür (und nur dafür) verwendet werden sollten. Im Einzelnen sind dies `<blockquote>` für längere Zitate, `<q>` für kürzere Zitate und `<cite>` für Quellenangaben. Zusätzlich kennt HTML noch das `cite`-Attribut (nicht zu verwechseln mit dem `cite`-Tag) über das Zitaten auch eine Quellenangabe zugewiesen werden kann.

Wie können Sie das testen?

Hier hilft leider mit den meisten Browsern nur der Blick in den Quelltext, da diese Elemente je nach Gestaltung nicht von herkömmlichem Text zu unterscheiden sind. Ausnahme: mit der oben beschriebenen Context Menu Extension bekommen Sie im Mozilla die Möglichkeit, zu den Quellen des Zitates zu springen, sofern welche per `cite="URL"` hinterlegt sind. Falls Sie Zitate mit Quellenangaben verwenden – der benötigte Menübefehl heißt "View Original Document of Quotation".

Ein Beispiel:

Im Einfach für Alle-Artikel '[Den Trend nicht verschlafen](#)' finden Sie folgende Textpassage:

*Die Zugänglichkeit im Internet regelt §11
Träger öffentlicher Gewalt ... gestalten ihre Internetauftritte ... schrittweise technisch so, dass sie von behinderten Menschen grundsätzlich uneingeschränkt genutzt werden können.*

Hier wurde das Zitat aus dem Gesetzestext mit dem für solche Zwecke vorgesehenen Tag `<q>` ausgezeichnet. Zusätzlich enthält das `<q>`-Tag noch ein `cite`-Attribut, in dem der [URL der Quelle](#) zum bequemen Nachlesen angegeben ist

In modernen grafischen Browsern kann man nun sogar noch per CSS die korrekten typografischen Anführungszeichen bestimmen:

```
q {
  quotes : ,\201c' ,\201d' ,\2018' ,\2019';
}
```

Tag 4: Sprachliche Besonderheiten

In der heutigen Folge geht es um eine der wenigen reinen Accessibility–Anforderungen der BITV: die Kennzeichnung von sprachlichen Besonderheiten. Ziel ist es, die Barrieren für Nutzer alternativer oder assistiver Technologien abzusenken und das allgemeine Textverständnis zu fördern.

Anforderung 4		Sprachliche Besonderheiten wie Wechsel der Sprache oder Abkürzungen sind erkennbar zu machen.
Bedingung 4.1	Prio. 1	Wechsel und Änderungen der vorherrschend verwendeten natürlichen Sprache sind kenntlich zu machen.

Was heißt das?

Sinn dieser Anforderung ist, dass Sprachausgaben, wie sie von Menschen mit Seh– oder Lernbehinderung, aber auch in mobilen Zugangsgaräten wie Autos genutzt werden, einen Sprachwechsel oder andere sprachliche Besonderheiten in einem Dokument erkennen. Sollten in einem hauptsächlich deutschen Text fremdsprachliche Worte vorkommen, so müssen diese auch als solche ausgezeichnet werden. Dazu gehören nicht nur mittlerweile fast eingedeutschte Begriffe wie Browser oder Website, sondern auch Anglizismen. Nur mit einer entsprechenden Kennzeichnung können Fremd– und Lehnwörter nach den Ausspracheregeln ihrer ursprünglichen Sprache vorgelesen werden.

Wie können Sie das testen?

Eine Anforderung, die nicht so einfach mit den Bordmitteln herkömmlicher Browser zu testen ist, es sei denn, man riskiert den Blick in den Quelltext. Erschwerend kommt hinzu, dass die meisten Browser (gerechnet in Marktanteilen) die in HTML vorgesehenen Konstrukte für die Anzeige des Sprachwechsels überhaupt nicht verstehen und auch assistive Technologien ihre lieben Nöte damit haben. So spricht der weit verbreitete Screenreader JAWS sämtliche Worte in der voreingestellten Sprache (also wahrscheinlich Deutsch) aus, egal ob der Autor die Sprachwechsel bei Fremdwörtern im Code deklariert hat oder nicht.

Uns ist bei unseren Tests bisher lediglich ein Sprachbrowser positiv aufgefallen, der diese Sprachwechsel beherrscht. Nur der IBM Home Page Reader spricht Lehn– und Fremdwörter dann nach den jeweiligen Ausspracheregeln aus, wenn diese mit dem dafür vorgesehenen Code:

```
<span lang="en"Browser</span,
```

ausgezeichnet sind. Dieser Sprachbrowser legt aber bei jedem Wechsel der Sprache eine deutlich spürbare Pause ein, was den Lese– oder besser Hörfluss nicht gerade erleichtert. Falls Sie sich Ihre Seiten mal anhören wollen, bei IBM gibt es eine 30–Tage [Testversion des Home Page Reader](#).

Ein Beispiel:

Bei der Überarbeitung der ‘Einfach für Alle’–Seiten wurde natürlich auch auf die Einhaltung dieser Anforderung geachtet. Zur Kontrolle der Sprachauszeichnungen wurde ein separates Style Sheet erstellt, in dem auf das Layout verzichtet wird und statt dessen alle sprachlichen Besonderheiten wie Sprachwechsel und Abkürzungen farbig hervorgehoben sind. Sie können [dieses Style Sheet](#) herunterladen und in jedem modernen Browser als ihr eigenes User Style Sheet installieren, das alle Angaben auf Webseiten überschreibt.

Bedingung 4.2	Prio. 2	Abkürzungen und Akronyme sind an der Stelle ihres ersten Auftretens im Inhalt zu erläutern und durch die hierfür vorgesehenen Elemente der verwendeten Markup-Sprache kenntlich zu machen.
---------------	---------	--

Was heißt das?

Eine schwierig umzusetzende Bedingung, und dies nicht nur, weil sie viel manuelle Nacharbeit bei der Einpflege der Inhalte bedeutet. Die eigentliche Schwierigkeit liegt eher darin, dass sich auch Sprachwissenschaftler nicht einig sind, was denn nun ein Akronym und was eine Abkürzung sei. Selbst das W3C benutzt in seiner Erklärung der Tags [falsche Beispiele](#).

Hintergrund dieser Bedingung ist die angenommene Fähigkeit, dass assistive Technologien dadurch in die Lage versetzt werden, Abkürzungen und Akronyme als Folge von Buchstaben wiederzugeben statt zu versuchen, diese als ein Wort auszusprechen. Nur gibt es hier in allen Sprachen mehr Ausnahmen als Regeln, z.B. ist UN ein Akronym wie GIF (ausgesprochen: giff oder dschiff) oder WAI (ausgesprochen: wäi) und müsste folgerichtig Un ausgesprochen werden – was natürlich keiner macht. Im Gegensatz dazu ist USA entgegen landläufiger Meinung kein aus den Anfangsbuchstaben gebildetes Akronym, sondern eine Abkürzung, da das O aus United States of America fehlt. Ganz unmöglich sind Abkürzungen, bei denen keine Einigkeit besteht, wie diese nun benutzt werden sollten: SQL kann sowohl Sequel als auch Es Kuh El ausgesprochen werden – für den Webentwickler ein nicht lösbares Problem.

Viele Browser, vor allem auch der Internet Explorer, verstehen auch in den aktuellsten Versionen immer noch nicht den `<acronym>`-Tag. Folgerichtig können viele assistive Technologien, die auf Browser aufsetzen, auch nichts mit der eingebauten Logik dieser Tags anfangen und müssen sich so auf ihre eingebauten Wörterbücher verlassen. Sogar Sprachbrowser wie der IBM Home Page Reader verstehen weder `<abbr>` noch `<acronym>`.

Angesichts dieses Durcheinanders hat das W3C im aktuellen [Entwurf für XHTML2](#) den `acronym`-Tag fallengelassen und sieht nur noch `<abbr>` vor. The `abbr` element indicates that a text fragment is an abbreviation (e.g., W3C, XML, Inc., Ltd., Mass., etc.); ü this includes acronyms.

Erschwerend kommt hinzu, dass nicht ein einziger User Agent die dafür vorgesehenen Hilfen zur Aussprache in CSS 2 (`speak: normal` oder `speak: spell-out`) unterstützt, auch die Screenreader und Sprachbrowser nicht. Was übrigens zur Folge hat, dass die entsprechenden Anweisungen aus Aural CSS in der jetzigen Form in Zukunft (also CSS 3) nicht mehr vorhanden sein werden.

Wie können Sie das testen?

Hier hilft mit den meisten Browsern nur der Blick in den Quelltext oder der oben beschriebene Weg über ein eigenes [User Style Sheet](#), dass die Tags, mit denen Abkürzungen und Akronyme ausgezeichnet wurden, farblich hervorhebt.

Moderne Browser wie Mozilla dekorieren Abkürzungen und Akronyme mit einem gepunkteten Unterstrich, sobald in die entsprechenden Tags ein zusätzliches `title`-Attribut eingefügt wurde. Nach der obigen Bedingung ist dies immer das erste Auftreten einer Abkürzung im Inhalt, was dem Besucher nicht so richtig viel nützt, wenn er von einem anderem Dokument über einen lokalen Anker an eine Textpassage weiter unten gekommen ist und so vielleicht die Erläuterung verpasst. Auch werden Menschen mit Lernbehinderungen, die unter Umständen bei jeder Instanz einer Abkürzung auf die Erklärung angewiesen sind, mit dieser Einschränkung nicht berücksichtigt.

Daher wird in der zukünftigen [WCAG 2](#) wohl allem Anschein nach auf diese Vorgabe verzichtet und stattdessen Verweise auf ein Glossar ermöglicht werden: „Expansion dictionaries, for instance in metadata, may be provided as an alternative to an expansion in the text of a document.“ Hier bei EfA beziehen wir uns der Einfachheit halber auf das [Glossar aus der BITV \(Anlage 2\)](#), das über das Tag

```
<link rel="glossary"
href="/artikel/bitv/bitv_glossar.html#inhalt"
title="Glossar" /
```

mit jeder Seite verknüpft ist.

Textbrowser wie Lynx zeigen diese *link relations* genannten Tags am Beginn eines Dokumentes an, aktuelle grafische Browser üblicherweise in einer zusätzlichen Toolbar.



Ein Beispiel:

Erstes Auftreten eines Akronyms und einer Abkürzung:

```
<acronym lang="en"
title="Cascading Style Sheets"CSS</acronym
<abbr title="und so weiter"usw.</abbr
```

Weitere Auftreten der Beiden:

```
<acronymCSS</acronym
<abbrusw.</abbr
```

Wem dies zu viel Handarbeit ist, dem seien der [Acrobot von accessify.com](#) oder das PHP-Skript bei [students.juniata.edu/gonzasio/temp/a.php](#) nahegelegt. Diese Tools dekorieren zumindest die gängigsten US-amerikanischen Abkürzungen und Akronyme mit dem notwendigen Code, so dass hier ein Teil der Handarbeit entfällt.

Bedingung 4.3	Prio. 2	Die vorherrschend verwendete natürliche Sprache ist durch die hierfür vorgesehenen Elemente der verwendeten Markup-Sprache kenntlich zu machen.
---------------	---------	---

Was heißt das?

Jedes gute HTML-Dokument gibt im Kopf an, in welcher Sprache der Text des Dokumentes verfasst wurde. Einer der esoterischeren Anwendungszwecke für diese Deklaration ist die Fähigkeit der Browser, je nach verwendeter Sprache die korrekten typografischen Anführungszeichen bei Zitaten zu setzen. Deutsche verwenden hier üblicherweise andere Zeichen als Franzosen und Amerikaner nehmen für alles das Zeichen für Zoll.

Wie können Sie das testen?

Auch hier hilft nur der Blick in den Quelltext. Wonach Sie suchen müssen, zeigt das folgende Beispiel.

Ein Beispiel:

In HTML 4 gibt es zu diesem Zweck das `lang`-Attribut des HTML-Tags, daher sollten Sie im Quelltext nach der Zeile:

```
<html lang="de"
```

bzw. bei XHTML-Dokumenten nach

```
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="de" lang="de"
```

suchen. Stehen diese am Anfang des Dokumentes unmittelbar nach der DTD, haben Sie diesen Test bestanden. Wenn dort nur <html steht, sollten Sie die Angabe der Sprache des Textes noch hinzufügen. Eine Liste aller von der ISO genormten Sprachkürzel finden Sie bei oasis-open.org.

Bedingung 11.4	Prio. 2	Der Nutzerin, dem Nutzer sind Informationen bereitzustellen, die es ihnen erlauben, Dokumente entsprechend ihren Vorgaben (z.B. Sprache) zu erhalten.
----------------	---------	---

Was heißt das?

Das Protokoll, dass die Unterhaltung zwischen Webbrowser und Webserver regelt, kennt eine ganze Liste von Möglichkeiten, mit denen ein Server die Fähigkeiten und Einstellungen eines Browsers abfragen kann. Eine dieser Abfragen ist der sogenannte [accept-language-Header](#), mit dem der Browser mitteilt, welche Sprachen sein Nutzer bevorzugt. Durch die Auswertung dieses Headers kann dann entschieden werden, ob von einem angeforderten Dokument vielleicht die passende Übersetzung vorliegt, die dann an Stelle des Originals ausgeliefert wird.

Wie können Sie das testen?

Wenn sie einen Webserver betreuen, der sich z.B. an Migranten mit geringen Deutschkenntnissen oder ausländische Studenten richtet, liegen vielleicht wichtige Dokumente in mehreren Sprachen vor. Sie können testen, ob die Übersetzungen korrekt angezeigt werden, indem Sie in Ihrem Browser die Spracheinstellungen auf die entsprechenden Ländercodes ändern. Im Internet Explorer geht dies z.B. über das den Befehl "Internetoptionen" im Menü "Extras". Hinter der Schaltfläche "Sprachen..." bekommen sie eine Liste angezeigt, in der wahrscheinlich nur "Deutsch (Deutschland) [de]" steht. Über "Hinzufügen" kommen Sie an eine Liste aller Sprachen von Afrikaans bis Zulu mit ihren entsprechenden Kürzeln. Wählen Sie hier nun z.B. "Griechisch [el]" oder "Türkisch [tr]" aus und definieren die neue Sprache über die Schaltfläche "Nach oben" als Ihre Bevorzugte.

Wenn nun die eventuell vorhandenen Übersetzungen an Stelle der Originale angezeigt werden, haben Sie den Test bestanden.

Ein Beispiel:

Das ganze gilt natürlich nur für den Fall, dass ein Dokument bereits zusätzlich in einer anderen Sprache vorliegt, deswegen müssen aber nicht gleich alle Websites übersetzt werden. Dies würde wohl auch nicht mehr in den Zuständigkeitsbereich der BITV fallen, da dies nichts mehr mit Barrierefreiheit im engeren Sinne zu tun hat. Sollte es allerdings irgendwann ein internationales Kürzel für die Deutsche Gebärdensprache geben, so würde auch hier die BITV wieder greifen und so die Kommunikation mit gehörlosen Menschen in diesem Bereich regeln.

Dieser Mechanismus kann aber nicht nur für große Dokumente eingesetzt werden, auch eher schlankere Sites wie Google setzen genau diesen Kunstgriff ein, um den Besuchern die ihren Einstellungen entsprechende Sprache anzuzeigen.

Tag 5: Tabellen

Lange genug die einzig verlässliche Methode, um Webseiten ein einigermaßen konsistentes Aussehen mitzugeben, haben Layout-Tabellen mittlerweile ihr Verfallsdatum deutlich überschritten. Folgerichtig beschreitet nun auch die BITV den Weg hin zu CSS und damit weg von Layouttabellen.

Anforderung 5	Tabellen sind mittels der vorgesehenen Elemente der verwendeten Markup-Sprache zu beschreiben und in der Regel nur zur Darstellung tabellarischer Daten zu verwenden.
----------------------	--

Eine recht schwammige Anforderung, die für sich alleine genommen genug Spielraum für Interpretationen lässt. Zum Glück wird diese Anforderung aber im Folgenden noch weiter detailliert, so dass auch hieraus eine brauchbare Handlungsanweisung werden kann. Die nächsten vier Anforderungen sind inhaltlich sehr eng verwandt, deswegen werden diese in einem Beispiel zusammen abgehandelt.

Bedingung 5.1	Prio.1	In Tabellen, die tabellarische Daten darstellen, sind die Zeilen- und Spaltenüberschriften mittels der vorgesehenen Elemente der verwendeten Markup-Sprache zu kennzeichnen.
Bedingung 5.2	Prio.1	Soweit Tabellen, die tabellarische Daten darstellen, zwei oder mehr Ebenen von Zeilen- und Spaltenüberschriften aufweisen, sind mittels der vorgesehenen Elemente der verwendeten Markup-Sprache Datenzellen und Überschriftenzellen einander zuzuordnen.
Bedingung 5.5	Prio.2	Für Tabellen sind unter Verwendung der hierfür vorgesehenen Elemente der genutzten Markup-Sprache Zusammenfassungen bereitzustellen.
Bedingung 5.6	Prio.2	Für Überschriftenzellen sind unter Verwendung der hierfür vorgesehenen Elemente der genutzten Markup-Sprache Abkürzungen bereitzustellen.

Was heißt das?

Wenn Ihre Seiten tabellarische Daten enthalten (also Wetterdaten, Börsenkurse oder Preislisten), stehen die Chancen nicht schlecht, dass es zu den verschiedenen Zeilen und Spalten auch die entsprechenden Zeilen- und Spaltenköpfe mit den Überschriften der einzelnen Datenzellen gibt. Für diesen Einsatzzweck sieht HTML die entsprechenden Tags wie `<caption>`, `<thead>` und `<tbody>` sowie `<th>` und `<td>` mit den Attributen `id` und `headers` vor. Ohne diese Tags und Attribute ist es für jeden nicht-grafischen Browser unmöglich, in komplexen Tabellen die Daten in Zeile 29, Spalte 5 noch den entsprechenden Überschriften der Spalte oder Zeile zuzuordnen.

Wie können Sie das testen?

Am besten geht dies natürlich mit einer Software, mit der die Zuordnung der Datenzellen zu den entsprechenden Überschriften auch tatsächlich funktioniert. Am [gestrigen Tag](#) hatten wir ja bereits den Sprachbrowser IBM Home Page Reader vorgestellt, der nun auch hier wieder zum Einsatz kommt.

Ein Beispiel:

Ein Beispiel für die Zuordnung der Datenzellen zu ihren Überschriften ist die Liste der bei EfA verwendeten Accesskeys in der Datei "[Hilfe](#)". Diese zugegebenermaßen sehr simple Tabelle besteht aus mehreren Teilen, nämlich der Beschreibung des Inhalts im Tag `<caption>` (Punkt 5.5 in der BITV), dem Kopf der Tabelle mit `<thead>` und `<th>` (BITV 5.1) sowie den Datenzellen im `<tbody>` mit `<td>`. Die Zuordnung

geschieht nun über die Attribute `id` für die jeweilige Überschrift und `headers` für die Datenzelle, die so die Verbindung zu ihrer Überschrift hält (BITV 5.2). Zusätzlich wurde mit dem `abbr`-Attribut noch genau der Text hinterlegt, der vor jeder Datenzelle ausgegeben werden soll (BITV 5.6). Dies kann eine Wiederholung der jeweiligen Überschrift sein, muss aber nicht, wie das Beispiel zeigt.

```
<table summary="Zusammenfassung">
<captionAccesskeys in der Navigation:</caption>
<thead>
<tr>
<th id="t" abbr="Taste">Taste</th>
<th id="b" abbr=": Link zu">Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td headers="t">0</td>
<td headers="b">Hauptmenü</span></td>
</tr>
[...]
```

Fertig gerendert und mit den entsprechenden Styles versehen sieht die Tabelle aus wie im folgenden Screenshot:

Accesskeys in der Navigation:	
Taste	Beschreibung
0	Hauptmenü
1	Anfang der jeweiligen Seite
2	Inhalt der Seite (Textbeginn)
3	Navigation der Seite
4	Inhaltsverzeichnis der Website
5	Suchfunktion

Vorgelesen im Tabellenmodus des IBM Home Page Reader hört sich der Table nun wie folgt an:

Accesskeys in der Navigation: Taste 0: Link zu Hauptmenü Taste 1: Link zu Anfang der jeweiligen Seite Taste 2: Link zu Inhalt der Seite (Textbeginn) Taste 3: Link zu Navigation der Seite Taste 4: Link zu Inhaltsverzeichnis der Website Taste 5: Link zu Suchfunktion

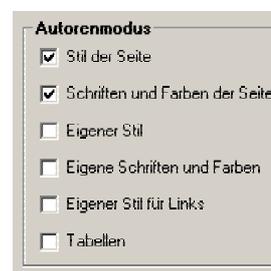
Bedingung 5.3	Prio.1	Tabellen sind nicht für die Text- und Bildgestaltung zu verwenden, soweit sie nicht auch in linearisierter Form dargestellt werden können.
Bedingung 10.3	Prio.2	Für alle Tabellen, die Text in parallelen Spalten mit Zeilenumbruch enthalten, ist alternativ linearer Text bereitzustellen.

Was heißt das?

Der Gnadestoß für [verschachtelte Layouttabellen](#).

Wie können Sie das testen?

Idealerweise testen Sie diese Anforderung mit einem [Textbrowser wie Lynx](#), der von Hause aus keine Tabellen versteht, oder einem Browser wie Opera, dem Sie die Darstellung von Tabellen abgewöhnen können. In Opera 6 (in der aktuellen Version 7.10 fehlt diese Checkbox, so dass Sie für diesen Test auf die ältere Version 6 ausweichen müssen) können Sie dies erreichen, indem Sie im Menü Datei / Einstellungen / Seitenlayout die Checkboxes "Tabellen" deaktivieren. Opera stellt nun eine linearisierte Fassung der Tabelleninhalte dar, d.h. die Inhalte werden in der Reihenfolge wiedergegeben, in der Sie im Quelltext stehen.



Die aktuellen Versionen von Opera gehen einen anderen Weg: hier gibt es die Möglichkeit, Tabellen über ein User Style Sheet zu deaktivieren. Der Effekt ist aber der gleiche wie bei den Vorgängerversionen – die Inhalte der Tabellen werden in ihrer Reihenfolge im Quelltext wiedergegeben.

Ein weiteres Tool zum Testen linearisierter Tabellen ist [Tablin](#) (kurz für: W3C WAI HTML Table Linearizer) des W3C.



Ein Beispiel:

Einfache, d.h. ein- bis zweispaltige und **nicht verschachtelte** Layout-Tabellen sind meistens kein Problem für die Barrierefreiheit, da diese auch von alternativen Ausgabemedien linearisiert dargestellt werden können.

Schwierig wird es, wenn Tabellen *mehrfach in einander verschachtelt* sind und so zum Beispiel die Überschrift eines Textes nicht unmittelbar vor dem dazu gehörigen Text kommt, sondern im Code am anderen Ende steht.

Insbesondere bei mehrspaltigem Satz kann dies dazu führen, dass zwei Überschriften, die in verschiedenen Zellen in einer Reihe nebeneinander stehen, zunächst auch nacheinander vorgelesen werden und danach dann die erste Spalte und dann die zweite Spalte, die aber zur zweiten Überschrift gehört.

Bedingung 5.4	Prio.1	Soweit Tabellen zur Text- und Bildgestaltung genutzt werden, sind keine der Strukturierung dienenden Elemente der verwendeten Markup-Sprache zur visuellen Formatierung zu verwenden.
---------------	--------	---

Was heißt das?

Wie wir soeben gesehen haben, sind Tabellen ja unter Umständen kein Hindernis auf dem Weg zur Barrierefreiheit, solange die Inhalte linearisiert wiedergegeben noch sinnvoll erscheinen. Sie sollten bei Layouttabellen, die rein der Gestaltung dienen, die im Beispiel zu den BITV-Punkten 5.1, 5.2 und 5.5 genannten Tags zur Strukturierung der Tabellen verwenden, da dies nur Verwirrung stiften würde. Natürlich mit Ausnahme von `<table>`, `<tr>` und `<td>`, aber *unter keinen Umständen* `<th>`, `<caption>` oder `<thead>`.

Wie können Sie das testen?

Jemand der Layouttabellen benutzt wird sich höchst wahrscheinlich nicht die Mühe machen, die oben genannten Tags einzusetzen. Daher können wir diesen Test getrost überspringen oder auf stichprobenartige Kontrollen des Quelltextes beschränken. Wenn Sie dabei in reinen Layouttabellen keine strukturierenden Elemente wie `<th>` oder `<thead>` finden, haben Sie den Test bestanden.

Tag 6: Fallback-Lösungen

Sie können nicht davon ausgehen, dass alle Benutzer den gleichen Browser wie Sie verwenden. Ein Blick in die Statistik Ihrer Website enthüllt die Vielzahl der unterschiedlichsten Browser aus verschiedenen Generationen, die auf Ihre Website zugreifen.

Der Blick in die Statistik verrät aber generell noch nichts über das tatsächliche Können und die Konfiguration der Browser, und die Zeiten, als die gesammelten HTML-Fähigkeiten aller bekannten Browser in eine [14 kB kleine Datei](#) passten, sind längst vorbei. Auch wird Ihnen das beste Statistikprogramm nicht verraten können, ob an einem Zielrechner lediglich ein grafischer Browser lief, oder ob vielleicht eine Braillezeile angeschlossen oder eine Sprachausgabe installiert war.

Anforderung 6	Internetangebote müssen auch dann nutzbar sein, wenn der verwendete Benutzer-agent neuere Technologien nicht unterstützt oder diese deaktiviert sind.
---------------	---

Dies betrifft nicht nur die Standardkonfigurationen der verschiedenen Browser, also die Frage, ob der Besucher überhaupt clientseitiges Java ausführen kann oder nicht, und deswegen unter Umständen ein Angebot ohne Alternativen nicht nutzen kann. Auch beim Einsatz von Plug-In-Formaten muss sichergestellt sein, dass Inhalte auch dann noch transportiert und Funktionen ausgeführt werden können, wenn das Endgerät diese Techniken nicht beherrscht.

Für manche Probleme gibt es allerdings keine Lösung. Wenn Ihr Angebot SSL zur Verschlüsselung von Transaktionen einsetzt, ist damit automatisch der Textbrowser Lynx ausgesperrt, der bis heute kein SSL beherrscht. Auf Verschlüsselung zu verzichten wäre ein nicht tragbarer Verlust für alle anderen Benutzer, und der Aufwand für ein paralleles, unverschlüsseltes System ist nicht zu rechtfertigen.

Bedingung 6.1	Prio.1	Es muss sichergestellt sein, dass mittels Markup-Sprachen geschaffene Dokumente verwendbar sind, wenn die zugeordneten Stylesheets deaktiviert sind.
---------------	--------	--

Was heißt das?

Wenn Sie in Ihren Seiten den Inhalt sauber von der Verpackung getrennt halten, sollte die Erfüllung dieser Bedingung eigentlich kein Problem sein. Allerdings bedingt der Einsatz von CSS, dass Ihre HTML-Dokumente bereits vorher schon sauber strukturiert sein müssen, um auch ohne CSS noch verwertbar zu sein.

Wie können Sie das testen?

Am [Tag 3](#) wurde ja bereits das [ToggleCSS-Favelet](#) vorgestellt, mit dem Sie sämtliche CSS-Formatierungen aus einer Seite entfernen. Laden Sie Ihre Seiten und entfernen Sie die Style Sheets. Noch einfacher geht es, wenn sie lokalen Zugriff auf die zu testenden Dateien haben: verschieben Sie einfach den Ordner, der die Style Sheets enthält. Üblicherweise ist dies das Verzeichnis /css/ in der obersten Ebene der Website. Wenn Sie nun die HTML-Dateien im Browser öffnen, werden die Styles nicht mehr gefunden und das reine HTML angezeigt.

Ein Beispiel:

Was alles schief gehen kann, wenn Inhalt, Funktion und Verpackung nicht sauber von einander getrennt wurden, sehen Sie bei der [Bundeswehr](#). Wenn man hier die Style Sheets deaktiviert, erhält man azurblaue Links auf dunkelblauem Hintergrund. Ursache hierfür ist, dass die Vordergrundfarbe der Links in einem externen Style Sheet deklariert wurde, die Hintergrundfarbe jedoch im HTML-Dokument selber.



Bedingung 6.2	Prio.1	Es muss sichergestellt sein, dass Äquivalente für dynamischen Inhalt aktualisiert werden, wenn sich der dynamische Inhalt ändert.
Bedingung 6.5	Prio.1	Dynamische Inhalte müssen zugänglich sein. Insoweit dies nur mit unverhältnismäßig hohem Aufwand zu realisieren ist, sind gleichwertige alternative Angebote unter Verzicht auf dynamische Inhalte bereitzustellen.

Was heißt das?

Auch wenn die [BITV](#) ausdrücklich alternative Textversionen als Sonderwege ausschließt, so gibt es doch manchmal keinen anderen Weg, um Inhalte für bestimmte Nutzergruppen zugänglich zu machen. Diese Bedingung stellt sicher, dass Textäquivalente genau so gepflegt und aktualisiert werden wie dynamische Inhalte.

Wenn Sie beim Einsatz von Java-Applets auf diesen Sonderweg verzichten wollen, empfiehlt sich die Lektüre der [IBM Guidelines for Writing Accessible Applications Using 100% Pure Java](#).

Ein Beispiel:

Es gibt viele Anwendungen im Netz, in denen z.B. ein Flash-Film oder ein Java-Applet Inhalte aus einer Datenbank bezieht und darstellt. Wenn diese Applets nun nicht zugänglich gestaltet werden können (was durchaus im Bereich des Wahrscheinlichen liegt), so sollten Sie

- a. eine statische Alternative im HTML-Format anbieten und
- b. dafür Sorge tragen, dass diese ebenso regelmäßige Updates der Inhalte erfährt wie die dynamische Version.

Wie können Sie das testen?

Diese Bedingung lässt sich natürlich nur mit einer manuellen Kontrolle der Inhalte überprüfen, dafür lässt sich dies aber mit fast jedem Browser bewerkstelligen. Idealerweise testen Sie dies mit einem Browser, der alle nötigen Plug-Ins und Java installiert hat (Java gehört mittlerweile bei vielen Browsern nicht mehr zum Lieferumfang, sondern muss separat installiert werden).

Bedingung 6.3	Prio.1	Es muss sichergestellt sein, dass mittels Markup-Sprachen geschaffene Dokumente verwendbar sind, wenn Scripts, Applets oder andere programmierte Objekte deaktiviert sind.
---------------	--------	--

Was heißt das?

Auch im aktuellsten Browser kann unter Umständen JavaScript abgeschaltet sein, um lästige Werbe-Pop-Ups zu verhindern. Beim Einsatz von Java, JavaScript und Flash sollte daher darauf geachtet werden, dass diese nur optional sind. Sie können sich beim Einsatz von Audio-Elementen in Flash auch nicht darauf verlassen, dass am Zielrechner überhaupt ein Lautsprecher vorhanden ist. Wenn Sie wichtige Inhalte mittels Audio transportieren, sollten Sie auf jeden Fall Textalternativen anbieten, um auch Menschen mit Hörbehinderungen den Zugang zu ermöglichen.

Wie können Sie das testen?

Auch diese Bedingung kann relativ einfach in den gängigen Browsern getestet werden. In den schon bekannten Schnelleinstellungen von Opera geht dies noch am einfachsten (F12). In anderen Browsern müssen Sie hierfür die Voreinstellungen oder Preferences ändern, daher beschränken wir uns auf Opera und seine Bordmittel. Im Internet Explorer ist der Knopf zu Ausschalten von JavaScript sogar so gut versteckt (und zudem noch nicht einmal mit JavaScript bezeichnet), dass ihn kaum jemand findet, daher lassen wir diesen Browser hier außen vor.



Deaktivieren Sie nun wieder alle optionalen Elemente (also JavaScript, Plug-Ins Audiosequenzen und Java) und testen Sie, ob noch alle Inhalte erreichbar und verständlich und alle Funktionen ausführbar sind. Dies gilt insbesondere für die Navigation und Formulare.

Ein Beispiel:

Viele Seiten im Netz benutzen DHTML-Menüs, die beim Überfahren mit der Maus ein Untermenü aufklappen. Um dies zu erreichen, ist normalerweise ein buntes Gemisch aus HTML, JavaScript und CSS nötig. Wenn nun JavaScript im Zielbrowser deaktiviert ist, klappt nichts auf und dem Benutzer bleiben wesentliche Teile der Navigation verborgen.

Dabei muss das Abschalten von JavaScript noch nicht mal eine bewusste Entscheidung des Benutzers sein – in vielen Behörden und Firmen werden Skripte aus Sicherheitsgründen erst gar nicht bis zum Benutzer gelassen, sondern [herausgefiltert](#).

Das heißt aber nicht, dass Sie auf diese Ausklappmenüs verzichten müssen, oftmals sind diese die beste Lösung, um komplexe Hierarchien abzubilden. Mittlerweile gibt es im Netz einige [gut dokumentierte Alternativlösungen](#), die auch mit abgeschaltetem JavaScript funktionieren oder gänzlich ohne auskommen. Links zu solchen Programmierbeispielen finden Sie im EfA-Artikel "[Barrierefrei zum Mitnehmen](#)"

Bedingung 6.4	Prio.1	Es muss sichergestellt sein, dass die Eingabebehandlung von Scripts, Applets oder anderen programmierten Objekten vom Eingabegerät unabhängig ist.
Bedingung 9.3	Prio.1	In Scripts sind logische anstelle von geräteabhängigen Event-Handlern zu spezifizieren.

Was heißt das?

Die übliche Art, Aktionen auf einer bereits geladenen Seite auszulösen ist über sogenannte Event Handler, die abfangen, wenn irgendetwas mit den Objekten einer Seite geschieht und dies an ein Skript oder ein Programm übergeben. Eine gängige Art, ein solches Event abzufangen ist z.B. mit `onMouseDown` – dies setzt aber, wie der Name schon sagt voraus, dass am Ausgabegerät tatsächlich eine Maus angeschlossen ist. Mit allen anderen Geräten und Zugangsarten, also auch rein tastaturbetriebenen Internetkiosks oder mobilen Endgeräten *ohne* Maus steht der Anwender dann im Regen.

Wie können Sie das testen?

Sie brauchen jetzt nicht gleich unter Ihren Schreibtisch zu kriechen und die Maus vom Rechner abzuklemmen, aber versuchen Sie doch einfach mal Ihre Website nur mit der Tastatur zu bedienen. Wenn Sie dann noch alle Funktionen ausführen können und alle Inhalte erreichen, haben Sie diesen Test bestanden.

Ein Beispiel:

Der Style Switcher, der hier bei Efa eingesetzt wird, um das Aussehen dieser Seiten verändern zu können, besteht aus einem Auswahlbox, aus der der entsprechende Stil ausgewählt werden kann.

```
<select onchange="setActiveStyleSheet(value);"
<option value="Efa House Style"
title="Gestaltung f&uuml;r grafische
Browser"Efa House Style</option
[...]
```

Die eigentliche Aktion wird durch das `onChange`-Event ausgelöst, also sobald der gewählte Eintrag in der Auswahlbox geändert und bestätigt wird. Dabei werden keine Angaben darüber gemacht, wie und warum sich etwas ändert, nur die Tatsache dass sich etwas ändert löst die Aktion aus.

Etwas aufwändiger wäre es, die Gestaltung über Textlinks geräteunabhängig zu ändern. Hier müssten tatsächlich mehrere mögliche Events abgefangen werden, wenn diese Funktion zum Beispiel auch über einen Accesskey erreichbar wäre:

```
onclick="setActiveStyleSheet(,foo');
onkeypress="setActiveStyleSheet(,foo');
```

Leider gibt es in HTML nicht für den gesamten Satz geräteabhängiger Event-Handler ein entsprechendes Äquivalent von logischen Event-Handlern. So nützt es bei `onDoubleClick` leider nichts, statt dessen zweimal die Aktion `onKeyPress` ausführen zu lassen.

Tag 7: Dynamik

Das Web ist weit mehr als nur ein Sammlung wohlstrukturierter statischer Texte. Animationen sind oft der einzige Weg, um manche Nutzergruppen zu erreichen oder um bestimmte Inhalte transportieren zu können. Aber gerade beim Einsatz dieser dynamischen Elemente braucht es besondere Sorgfalt, um nicht hinterher mit einer komplett unzugänglichen Site da zu stehen.

Anforderung 7	Zeitgesteuerte Änderungen des Inhalts müssen durch die Nutzerin, den Nutzer kontrollierbar sein.
---------------	--

Eine Anforderung, die sich auf die Bedürfnisse vieler unterschiedlicher Nutzergruppen bezieht. Mit Zeitgesteuerte Änderungen des Inhalts können sowohl automatische Weiterleitungen in einer Seite als auch flackernde oder blinkende Inhalte sein. Auch Animationen, sei es per DHTML oder Flash fallen in diese Kategorie. Aber auch das gute alte `blink`-Tag kommt hier zu neuen Ehren.

Bedingung 7.1	Prio.1	Bildschirmflackern ist zu vermeiden.
Bedingung 7.2	Prio.1	Blinkender Inhalt ist zu vermeiden.

Was heißt das?

Bei Menschen mit Epilepsie können in bestimmten Frequenzen flackernde Inhalte einen Anfall auslösen. Daher sollten Sie schnelle Animationen oder zu schnelle Wechsel von Hell und Dunkel vermeiden. Dies gilt sowohl für animierte GIFs als auch alle sonstigen eingebetteten Formate wie Flash oder Java.

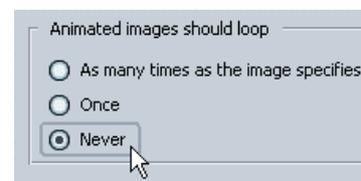
Wie können Sie das testen?

Hier hilft tatsächlich nur die visuelle Überprüfung, d.h. Sie müssen sich Ihre Website ansehen und auf blinkende Inhalte untersuchen. Im MS IE können sie die Animation von GIFs ganz einfach abschalten, in dem Sie das folgende Bookmarklet in die Links-Symbolleiste (Menü: Ansicht / Symbolleisten / Links) ziehen:

[Animierte GIFs aus](#)

Ziehen Sie den obigen Link einfach auf die Symbolleiste des Internet Explorers. An der Einfügemarke bekommen Sie nun einen Button, mit dem sie die Animationen stoppen können.

In vielen Browsern können die Animation von GIFs auch über die Voreinstellungen unterbunden werden, so in den aktuellen Mozillas über den gut versteckten Dialog Edit / Preferences / Privacy & Security / Images /.



Ein Beispiel:

Blinkende Inhalte lenken übrigens auch vom Lesen eines Textes ab, so dass deren Vermeidung allen Benutzern zu Gute kommt. Der Mensch war die längste Zeit seiner Geschichte auch ein Beutetier und ist heute noch darauf "programmiert". Die Erklärung, warum das Auge immer zu Animationen in das sekundäre Gesichtsfeld gezogen wird ist einfach: Bewegungen, die außerhalb des primären Blickfeldes

geschehen, könnten ja von einem Säbelzahn tiger kommen, der dort im Gebüsch sitzt und auf Beute wartet.

Aber auch ohne Flash oder animierte GIFs können Sie Inhalte ihrer Seite rhythmisch blinken lassen: das Netscape-eigene `blink`-Tag und dessen Pendant aus CSS (`text-decoration: blink;`) fallen ebenso unter dieses Verbot.

Wenn Sie wichtige Informationen zu Ihrem Webangebot in einem animierten GIF ablegen, so sollten diese bereits im ersten Frame der Animation stehen. Diesen Frame können Sie natürlich so anlegen, dass er nur einige Millisekunden angezeigt wird, und danach dann die eigentliche Animation kommt. So verhindern Sie, dass Benutzern, die Animationen unterbunden haben, wichtige Informationen vorenthalten werden.

Bedingung 7.3	Prio.1	Bewegung in mittels Markup-Sprachen geschaffener Dokumente ist entweder zu vermeiden oder es sind Mechanismen bereitzustellen, die der Nutzerin, dem Nutzer das Einfrieren der Bewegung oder die Änderung des Inhalts ermöglichen.
---------------	--------	--

Was heißt das?

Gerade bei Animationen, in denen sich größere Mengen an Inhalten befinden, können Sie nicht davon ausgehen, dass diese von allen Besuchern in der gleichen Geschwindigkeit verarbeitet werden können. Dazu gehören Menschen mit Leseschwäche oder Lernbehinderung, aber auch Menschen, die der deutschen Sprache vielleicht nicht so mächtig sind wie ein Muttersprachler und daher länger zum Lesen oder Wahrnehmen der Informationen brauchen.

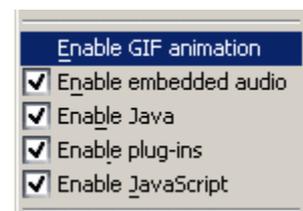
Bei Flash können Sie sich jetzt entspannt zurücklehnen und die Lösung des Problems auf den Benutzer abwälzen, denn dieses Programm bietet seit jeher die Möglichkeit, über ein Kontextmenü Animationen anzuhalten, vor- oder zurückzuspulen. Bei Animationen in Java-Applets sollten Sie für den Benutzer eine gleichwertige Funktion bereitstellen, mit der dieser die Kontrolle über Ablauf und Geschwindigkeit der Animation erhält. Beim [Java Accessibility Program](#) der University of Wisconsin finden Sie ein [Programmierbeispiel](#), das zeigt, wie der Nutzer Bewegungen abschalten kann.

Wie können Sie das testen?

Bei Animationen mit DHTML ist sowohl der Test als auch die Bedingung selbst einigermaßen schwer umzusetzen. Wenn Sie die übliche Kombination aus HTML, CSS und JavaScript einsetzen, um Inhalte über den Bildschirm zu bewegen, hilft nur das Abschalten von JavaScript, um diese Bewegung zu verhindern.

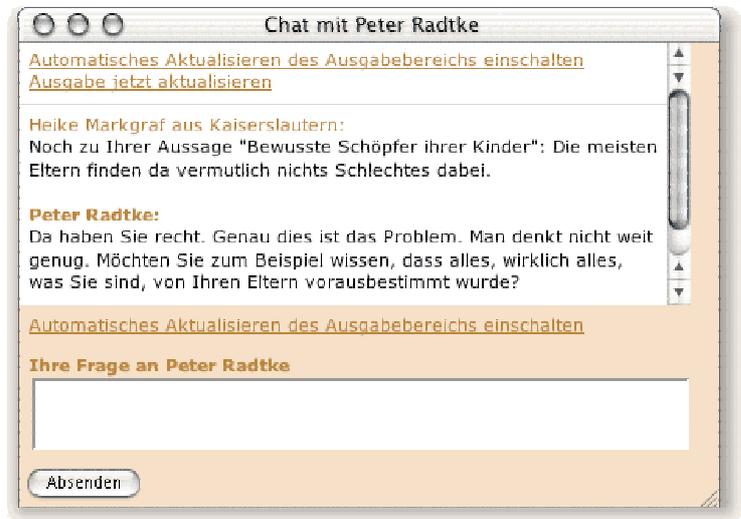
Auch animierte GIFs können nicht durch Interaktion mit dem Objekt selbst, sondern nur durch einen Eingriff in die Voreinstellungen des Browsers (oder besser gesagt der meisten Browser) gestoppt werden.

In Opera helfen hier wieder die bereits bekannten Schnelleinstellungen (F12), in denen animierte GIFs verhindert werden können, andere Browser bieten vergleichbare Knöpfe in ihren allgemeinen Einstellungen.



Ein Beispiel:

Problematisch sind auch Echtzeit-Chats, in denen sich üblicherweise das Ausgabefenster in bestimmten Intervallen selbsttätig aktualisiert. Diese Chats können auf verschiedenen Client-seitigen Techniken wie Java basieren, oder auch komplett in reinem HTML ablaufen. Die Aktion Mensch nutzt ein solches System im Rahmen Ihrer Ethik-Kampagne unter 1000fragen.de, wo in regelmäßigen Abständen Besucher- und Experten-Chats angeboten werden.



Die nach intensiven Tests hierfür gefundene beste Lösung ist ein rein HTML-basiertes System, in dem der Besucher selbst zwischen automatischer und manueller Aktualisierung wählen kann.

Bedingung 7.4	Prio.1	Automatische periodische Aktualisierungen in mittels Markup-Sprachen geschaffener Dokumente sind zu vermeiden.
---------------	--------	--

Was heißt das?

Zum Beispiel die vor einigen Jahren in Mode gekommenen Newsticker, die wie ein vertikales oder horizontales Laufband Nachrichtenschnipsel über den Bildschirm schieben. Dazu können Java-Applets oder JavaScripts verwendet werden, oder der MS IE-eigene `<marquee>`-Tag. Auf letzteren sollten Sie allein schon deswegen verzichten, weil dieser überhaupt nur vom Internet Explorer verstanden wird und auch nie Teil des HTML-Standards war.

Wie können Sie das testen?

Beim `<marquee>`-Tag ist dieser Test relativ einfach: Nehmen Sie einfach einen anderen Browser als den Internet Explorer und versuchen Sie dann, den Inhalt Ihres Newstickers zu erfassen. Bei anderen Technologien, mit denen sich periodische Aktualisierungen schaffen lassen, helfen wieder die Schnelleinstellungen des Opera (F12): Schalten Sie alle optionalen Inhalte ab, also Plug-Ins, Java und JavaScript.

Periodische Aktualisierungen können aber auch durch sogenannte *meta-refreshes* in Frames oder eingebetteten Frames (`iframes`) erreicht werden. Um diese Technik kümmert sich die nächste Bedingung.

Bedingung 7.5	Prio.1	Die Verwendung von Elementen der Markup-Sprache zur automatischen Weiterleitung ist zu vermeiden. Insofern auf eine automatische Weiterleitung nicht verzichtet werden kann, ist der Server entsprechend zu konfigurieren.
---------------	--------	--

Was heißt das?

Die weit verbreitete Technik, `meta-refreshes` zur Weiterleitung einzusetzen, ist für viele Benutzer verwirrend und kann gerade bei der Verwendung assistiver Technologien desorientierend wirken.

Wie können Sie das testen?

Wenn Sie bei der Überprüfung Ihrer Seiten auf Texte wie diesen stoßen:

„Wenn Ihr Browser automatische Weiterleitung unterstützt, werden Sie in 5 Sekunden weitergeleitet. Ansonsten klicken Sie hier <http://www.servername.de/seite>“

dann setzt Ihr Webangebot genau diese Technik ein. Das dafür verantwortliche Tag steht im Kopf Ihres Dokumentes und sieht in etwa wie folgt aus:

```
<meta http-equiv="refresh"
content="5; http://www.servername.de/seite"
```

Ein Beispiel:

Wenn Inhalte eines Webauftrittes verschoben werden, kann es passieren, dass auf sie zeigende Hyperlinks ins Leere laufen. Der Server kann aber entsprechend konfiguriert werden, dass automatisch serverseitig auf den neuen URL der Datei weitergeleitet wird. In der Fachsprache der Serveradministratoren ist dies ein "[301 moved permanently](#)". Dies hat zudem den Vorteil, dass hierdurch der Datenverkehr zwischen Browser und Server reduziert wird und Ihre Seiten insgesamt performanter werden.

Beim Relaunch der ‘Einfach für alle’-Seiten im Mai 2003 wurden viele Inhalte überarbeitet und die gesamte Struktur der Website geändert. Wie immer bei einem solchen Umbau stellte sich das Problem, dass Nutzer mit alten Bookmarks oder Links, die von außerhalb auf das Webangebot zeigen, ins Leere laufen und einen der berüchtigten [Fehler 404](#) zurückbekommen. Daher wurde die `.htaccess`-Datei des Servers so eingerichtet, dass sie diese Anfragen automatisch an den neuen Speicherplatz (URL) der Dokumente weitergibt.

Tag 8: Zugang

In dieser Folge geht es um die Zugänglichkeit von Inhalten mit eingebetteten Benutzerschnittstellen – gemeint sind programmierte Objekte, die über das bloße HTML oder Bilder hinausgehen und eigene Schnittstellen zur Ausgabe und Bedienung zur Verfügung stellen.

Anforderung 8		Die direkte Zugänglichkeit der in Internetangeboten eingebetteten Benutzerschnittstellen ist sicherzustellen.
Bedingung 8.1	Prio. 1	Programmierte Elemente (insbesondere Scripts und Applets) sind so zu gestalten, dass sie entweder direkt zugänglich oder kompatibel mit assistiven Technologien sind.

Was heißt das?

Im Artikel zur [BITV-Anforderung 1](#) wurden ja bereits die alternative Inhalte zu eingebetteten statischen Objekten diskutiert. Diese spezielle Anforderung 8 kümmert sich nun um die geräteunabhängige Zugänglichkeit von Inhalten, die üblicherweise per `object`-Tag in eine Seite eingebunden werden oder ein zusätzliches Plug-In benötigen, um im Browser ausgeführt zu werden.

Nach dieser Anforderung müssen **sämtliche** Inhalte einer Web-Angebotes genauso zugänglich sein wie der Rest der Seite auch, egal welche Technologie eingesetzt wird. Folgerichtig gelten also für Flash- und Java-Anwendungen oder auch für eingebettete oder verlinkte PDF-Dokumente die gleichen Kriterien für die Ausgabe in alternativen oder assistiven Medien und die Bedienung per Tastatur oder anderen Hilfsmitteln.

Wie können Sie das testen?

Die direkte Zugänglichkeit und Bedienbarkeit von programmierten Elementen wurde bereits am [Tag 6 dieser Serie](#) besprochen. Die hier ausdrücklich erwähnten assistiven Technologien, die z.B. von Menschen mit Sehbehinderung, motorischer Behinderung oder Lernbehinderung benutzt werden, sind sehr vielfältig, aber auf Grund der teilweise immensen Kosten nicht unbedingt immer auf dem neuesten Stand der Technik.

Daher können Sie ohne Tests mit tatsächlichen Benutzern solcher Technologien nicht mit Sicherheit davon ausgehen, dass Ihre Seiten auch wirklich mit einer Vielzahl von Hilfsmitteln bedienbar sind. Falls Sie [Kontakt zu Nutzern assistiver Technologien](#) wie Screenreadern suchen, gibt es hierfür Institutionen wie [Web for All](#) in Heidelberg, die über sehr erfahrene Tester verfügen. Eine weitere gute Möglichkeit zum Austausch mit betroffenen Experten sind Mailinglisten wie die [W3C-WAI-DE-Liste](#) des deutschen W3C-Büros am Fraunhofer-Institut.

Bedingung 9.2	Prio. 1	Jedes über eine eigene Schnittstelle verfügende Element muss in geräteunabhängiger Weise bedient werden können.
---------------	---------	---

Was heißt das?

Eingebettete Inhalte benötigen in den meisten Fällen Zusatzmodule (sogenannte Plug-Ins wie zum Beispiel Acrobat oder Flash), die dem Browser die Verarbeitung dieses Formats überhaupt erst ermöglichen. Diese Formate werden üblicherweise nicht im HTML-Quellcode, sondern mit zusätzlichen Programmen

erstellt. Daher verweisen die der BITV zu Grunde liegenden [Web Content Accessibility Guidelines \(WCAG\)](#) an dieser Stelle auf eine andere Empfehlung des W3C, die [Authoring Tool Accessibility Guidelines \(ATAG\)](#). In den ATAG-Richtlinien ist festgelegt, dass auch diese Programme den Autor bei der Erstellung barrierefreier Inhalte unterstützen müssen.

Hier beginnen nun die Probleme für Autoren, wenn zum Beispiel Inhalte im PDF-Format angeboten werden sollen. Für die Version 4 des Acrobat-Plug-Ins existiert zwar ein Accessibility-Plug-In, das zwar mit vielen assistiven Technologien prinzipiell kompatibel ist, aber je nach Aufbau des PDF-Dokumentes insbesondere bei Formularen oder Spaltensatz eine absolut unzugängliche Ausgabe der Inhalte produziert. Die Version 5 hat hier einiges an Verbesserung gebracht, nur ist diese wiederum noch so neu, dass die wenigsten assistiven Technologien damit überhaupt umgehen können. Erst die kommende Version 6 verspricht hier Abhilfe, da der Hersteller Adobe dem Vernehmen nach einen Screenreader in das Plug-In integrieren wird.

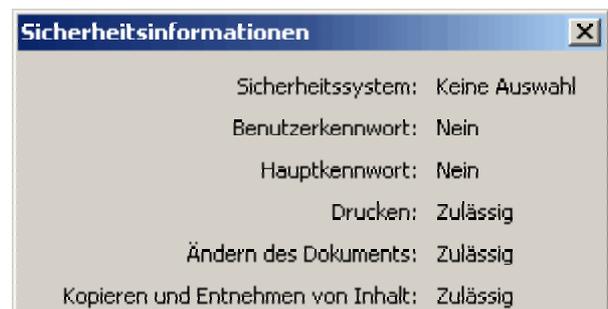
Ein Beispiel:

Ein Kardinalfehler, der immer wieder bei der Erstellung von PDF-Dokumenten zu beobachten ist, passiert mit dem meist unbewussten Ausschluß von Screenreadern über die Sicherheitseinstellungen des Dokumentes. In diesen Sicherheitseinstellungen kann der Ersteller des PDF-Dokumentes festlegen, ob ein Kopieren der Inhalte erlaubt ist oder nicht.

Diese Vorgabe wird oftmals benutzt, um eine scheinbare Sicherheit gegen die unerlaubte Vervielfältigung oder Veränderung des Dokumentes zu erreichen. Allerdings sind die gängigen Screenreader genau auf diese Funktion des Kopierens angewiesen, um überhaupt an die Inhalte des Dokumentes zu kommen und diese wiedergeben zu können.

Wie können Sie das testen?

Sie finden die Einstellungen für Ihre PDF-Dokumente im Menü Datei / Sicherheitsinformationen des Acrobat Reader. Wenn in diesem Dialog der Eintrag 'Kopieren und Entnehmen von Inhalt' auf 'Nicht zulässig' steht, dann verhindert dies auch das Entnehmen von Inhalt durch einen Screenreader und ist somit nicht mehr zugänglich.



Wenn Sie trotz bester Bemühungen keine barrierefreie Version Ihrer PDF-Dokumente erzeugen können, sollten Sie im Interesse Ihrer Benutzer zusätzlich alternative Formate wie RTF oder ASCII-Text anbieten, die mit allen gängigen Textverarbeitungen verarbeitet werden können.

Noch ein Beispiel:

Etwas schwieriger ist diese Vorgabe der BITV bei eingebetteten Formaten wie Flash umzusetzen. Die aktuelle Version des Flash Players 6 unterstützt zwar unter Windows die MSAA-Schnittstelle, über die geeignete Screenreader an die Textinhalte eines Flash-Films gelangen können, nur wird diese Schnittstelle noch lange nicht von allen Screenreadern verstanden und benutzt.



Mittlerweile können in Flash MX zwar alternative Texte zu grafischen Buttons definiert werden, aber bei den meisten Dingen bleiben die Accessibility-Features von Flash weit hinter den Möglichkeiten von HTML 4 zurück. Problematisch sind insbesondere Formulare in Flash, da Beschriftungen (Labels) den Formularelementen nicht sinnvoll zugeordnet werden können.

Die Erreichbarkeit der Formularelemente in einem Flash-Film ist somit auch für geübte Nutzer von Sprachausgaben kaum zu bewältigen. Erschwerend kommt hinzu, dass sämtliche Browser bis zur 5er Generation Probleme mit dem für Flash-Filme nötigen `object`-Tag haben. In manchen älteren Browsern ist es vollkommen unmöglich, zu einem per `object` eingebetteten Inhalt per Tabulator-Taste oder Accesskey zu navigieren; in manchen neueren Browsern ist dies nun möglich, aber dafür kommt man dann oftmals per Tabulator-Taste nicht mehr aus dem Objekt hinaus in die eigentliche HTML-Seite

Wenn Ihre Flash-Filme also Formulare beinhalten, bleibt Ihnen beim jetzigen Stand der Technik nur der Ausweg über ein zusätzliches alternatives Angebot in klassischem HTML: „If the interface of the embedded object cannot be made accessible, an alternative accessible solution must be provided“.

Tag 9: Unabhängigkeit

Wenn Ihre Seiten für alternative Ausgabegeräte (assistive Technologien wie Sprachausgaben, Braillezeile oder Joystick) zugänglich sind, werden sie auch mit Kiosksystemen ohne Tastatur oder den Hilfsmittel von Menschen mit motorischer Behinderung bedienbar sein.

Anforderung 9	Internetangebote sind so zu gestalten, dass Funktionen unabhängig vom Eingabegerät oder Ausgabegerät nutzbar sind.
---------------	--

Was heißt das?

Bei oberflächlicher Betrachtung ist dies eine sehr allgemeine Formulierung, die als Fallback für alle Eventualitäten dient, die nicht explizit in der BITV erwähnt sind. Wenn man allerdings bedenkt, dass Internet nicht mehr ausschließlich auf großen Computermonitoren stattfindet, wird auch der eigentliche Sinn dieser Vorgabe klar: Nicht jedes Ausgabegerät verfügt über die selben Möglichkeiten wie der Rechner, auf dem eine Seite entwickelt wurde und auch ein Drucker ist ein mögliches Ausgabegerät.

Wenn Ihre Seiten aus sauberem HTML 4 oder XHTML bestehen, sind sie damit schon fast automatisch unabhängig vom Ausgabegerät. HTML 4 verzichtet auf viele Elemente, die in den Vorgängerversion 3.2 rein zur grafischen Präsentation gedacht waren. Weniger ist hier also mehr, vor allem dann, wenn Sie die komplette Gestaltung in CSS auslagern. Allerdings sollten auch Server, serverseitige Anwendungen und Skripte darauf eingerichtet sein, den Inhalt an beliebige Ausgabeformen auszuliefern, wie nachfolgender Screenshot illustriert.

N.B.: bei dem eingesetzten Browser handelt es sich um iCab für den Apple Macintosh. Dieser Browser ist bis heute als einziger in der Lage, den kompletten HTML 4–Satz korrekt anzuwenden. Man kann diesem Browser allerdings auch eine eigene, frei erfundene Identifikation mitgeben, mit der dieser sich bei einem Webserver meldet.

Im konkreten Fall war dies:

"IFA Trabant 1.1 (compatible; VEB Sachsenring; Zweitakt 1:25)".

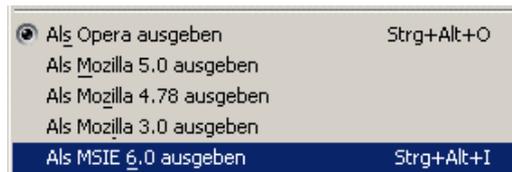
Sie lachen? Was wären denn, wenn sich ein 7er BMW nach dem Fahrplan für den Autoreisezug hätte erkundigen wollen?



Wie können Sie das testen?

Bei diesem Test brauchen wir wieder einen Opera, der ein wenig umkonfiguriert wird. Viele Websites arbeiten mit Browserabfragen, die nach bestimmten Mustern suchen, mit denen sich Browser beim Server zu erkennen geben. Je nach dem, was der Browser auf die Abfrage antwortet, kann es passieren, dass ihm dann ein anderer Inhalt serviert wird, als dies bei einer anderen Antwort der Fall gewesen wäre. Dieser Inhalt sollte natürlich eigentlich identisch sein, was aber oftmals nicht der Fall ist.

In Opera können Sie dies testen, in dem Sie die Browser-identifikation einfach auf Opera oder einen anderen Browser stellen (Schnelleinstellungen mit F12) und kontrollieren, ob Sie immer noch an die selben Inhalte und Funktionen kommen.



Ein anderer wichtiger Test ist es, sich die eigene Website mal mit unterschiedlich großen Browserfenstern anzusehen und dann zu überprüfen, ob die Gebrauchstauglichkeit immer noch vorhanden ist. Opera bietet hierfür die Möglichkeit, die aktuelle Fenstergröße anzuzeigen. Die entsprechende



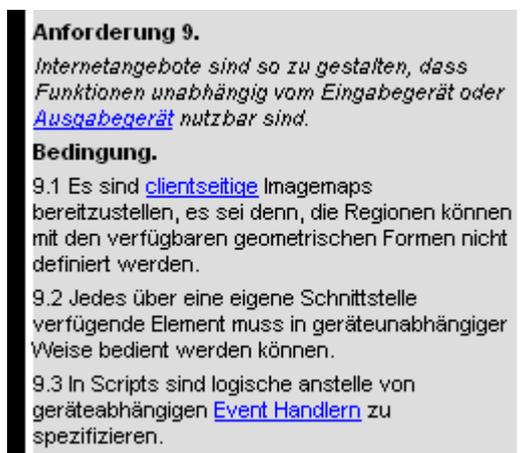
Einstellung finden Sie in den Einstellungen unter Fenster für Webseiten / Fenstergröße anzeigen.

Danach zeigt Opera die aktuelle Fenstergröße zusammen mit dem Titel der Seite im Reiter an. Schieben Sie das Fenster einfach mal auf 640x480 zusammen und testen Sie, ob Ihr Layout dann noch stimmt und die Seiten nach wie vor ohne zusätzliche Eingriffe (horizontaler Scrollbalken!) zu bedienen sind.

Wenn Sie die Befürchtung hegen, dass Ihre Nutzer auch mit mobilen Endgeräten wie Palm oder PocketPC auf Ihre Seiten zugreifen könnten, sollten Sie das Fenster ruhig noch weiter verkleinern: die aktuellen Palms haben ein Display von 320x320 Pixel. Für Ihre Website bleibt allerdings noch weniger übrig, da Sie von dieser Fläche noch die Menüleisten und Toolbars sowie den unweigerlich auftretenden Rollbalken abziehen müssen.

Der PocketPC stellt sogar nur 320x240 Pixel dar, immerhin noch ein Viertel eines 14"-Monitors. Abzüglich der Menüleiste (-26px vertikal) und der Captionbar (-26px vertikal) bleiben hier nur noch maximal 268 Pixel übrig. Von der Horizontalen gehen dann noch 11 Pixel für den Scrollbalken verloren, so dass man insgesamt nur noch auf eine Fläche von 268x229 Pixel kommt. Die Tests für den PocketPC Internet Explorer sollten Sie übrigens mit abgeschaltetem CSS vornehmen, da der diese Sprache ohnehin nicht versteht.

Auch hier hat Opera wieder einen passenden Menübefehl parat: in den aktuellen 7er-Versionen finden Sie im Menü "View" den Befehl "Small Screen" (N.B. zum Zeitpunkt der Veröffentlichung war leider nur eine englische Vorabfassung verfügbar, so dass wir hier leider die englischen Menübefehle benutzen müssen.) Mit diesem Befehl simuliert der Desktop-Opera das Aussehen seines Pendant für mobile Endgeräte.



Die Unabhängigkeit vom Eingabegerät ist schon etwas aufwändiger zu bewerkstelligen, da hierfür zusätzliche Tags und Attribute in die Dokumente eingebaut werden müssen, um die sich die folgenden Bedingungen 9.4 und 9.4 kümmern.

Falls hier nach den Bedingungen 9.1 bis 9.3 suchen, diese sind wegen ihrer inhaltlichen Nähe bereits weiter vorne in dieser Serie abgehandelt worden. 9.1 finden Sie an [Tag 1 bei 1.5](#), 9.2 an [Tag 8 bei 8.1](#) und 9.3 an [Tag 6 bei 6.4](#).

Bedingung 9.4	Prio. 2	Es ist eine mit der Tabulatortaste navigierbare, nachvollziehbare und schlüssige Reihenfolge von Hyperlinks, Formularkontrollelementen und Objekten festzulegen.
---------------	---------	--

Was heißt das?

Eines der Elemente für die Bedienung mit alternativen Eingabegeräten ist das `tabindex`-Attribut, mit dem die geforderte Reihenfolge festgelegt wird, in der sich ein Nutzer mit der Tabulatortaste durch eine Seite bewegen kann. Dies heißt natürlich nicht, dass Sie nun alle erdenklichen Elemente einer Seite mit einem solchen `tabindex` ausstatten sollten. Die weitaus meisten Browser gehen ohne das Vorhandensein von `tabindex` die anklickbaren Objekte einer Seite in der Reihenfolge durch, in der Sie im Quelltext stehen. Erst wenn diese Reihenfolge z.B. bei einer Transaktionsseite nicht mehr sinnvoll für die Bewältigung der Aufgabe erscheint, sollten Sie dies mit `tabindex` korrigieren.

Wenn Ihre Seite in größere logische Blöcke unterteilt ist, können Sie das `tabindex`-Attribut auch benutzen, um zwischen diesen Blöcken zu navigieren. Dabei müssen diese nicht fortlaufend nummeriert sein, sie können zum Beispiel auch in einem Dokument mit 1,2,3 und 4 anfangen und dann mit 10, 11, 12,... weitermachen. So bleibt Ihnen Raum für eventuelle Erweiterungen, bei denen Sie nicht immer wieder alles neu nummerieren müssen.

Wie können Sie das testen?

Lassen Sie die Maus ruhen und benutzen Sie nur die Tabulator- und Eingabetasten zur Bewegung durch Ihre Website. Wenn Sie nun alle Inhalte in einer Reihenfolge erreichen, die sinnvoll und nachvollziehbar erscheint, haben Sie diesen Test bestanden.

Bedingung 9.5	Prio. 2	Es sind Tastaturkurzbefehle für Hyperlinks, die für das Verständnis des Angebots von entscheidender Bedeutung sind (einschließlich solcher in clientseitigen Imagemaps), Formularkontrollelemente und Gruppen von Formularkontrollelementen bereitzustellen.
---------------	---------	--

Was heißt das?

Die hier geforderten Tastaturkurzbefehle heißen in HTML "Accesskeys" und funktionieren im Grunde genommen wie die Shortcuts zu den Menübefehlen in den unterschiedlichen Betriebssystemen. Alt-D öffnet unter Windows üblicherweise das "Datei"-Menü, Alt-B steht für Bearbeiten und so weiter.

Und genau da beginnt das Problem: leider sind die weitaus meisten Buchstaben schon in den gängigen Browsern mit irgendwelchen Menübefehlen belegt, so dass fast nur noch die Ziffern des 10er-Blocks für die Inhalte und Funktionen einer Webseite übrig bleiben. Selbst die Zahlen werden von manchen assistiven Tools für Sonderfunktionen verwendet, so wie Alt+1 vom IBM Home Page Reader für dessen Überschriftenfunktion.

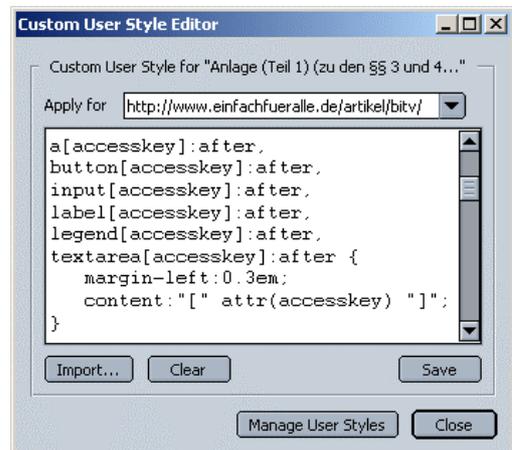
Der Experte Jan Eric Hellbusch hat dieses Problem auf seiner Website im Artikel [Navigieren ohne Maus](#) zusammengefasst und weist dort auch den [hier bei EFA verwendeten Lösungsansatz](#) auf.

Bei komplexen Formularen mit vielen Kontrollelementen ist es also unmöglich, für alle einen passenden Accesskey zu finden, der nicht irgendwo schon mit einer anderen Funktion belegt ist. Daher bleibt nur die Empfehlung, sich auf die maximal zehn zur Verfügung stehenden Zahlen zu beschränken und diese möglichst sinnvoll zu verteilen. Wie dies geschieht, ist natürlich dem Seitenbauer selbst überlassen, da es

hierfür keinen einheitlichen Standard gibt. Sie sollten die Verwendung der Accesskeys an geeigneter Stelle dokumentieren. Hier bei EfA geschieht dies zum Beispiel über den Aufruf der Datei [Hilfe](#).

Wie können Sie das testen?

Generell geben die heutigen grafischen Browser einen hinterlegten Accesskey nicht für den Benutzer sichtbar aus (macht aber nichts, der Textbrowser Lynx oder assistive Tools wie Webformator und Home Page Reader machen das auch nicht), daher hilft hier nur ein kleiner Kniff, um den Blick in den Quelltext zu vermeiden. Sie können in einem der modernen Browser wie Opera oder Mozilla ein Benutzer-Style Sheet anlegen, das eventuell vorhandene Accesskeys in die Dokumente reinschreibt.



Fügen Sie einfach hierzu die Zeile

```
*[accesskey]:after {  
  content: „[" attr(accesskey) „]“;}
```

ein und schon werden Ihnen die Accesskeys angezeigt. Dieser verwendete CSS-Trick nennt sich "Generated Content" und kann zu vielen verschiedenen Dingen benutzt werden, die das Surfen oder Testen erleichtern. Wir haben einige davon zusammengestellt und in der Datei [debug.css](#) zum Download bereitgestellt.

Tag 10: Kompatibilität mit assistiven Technologien

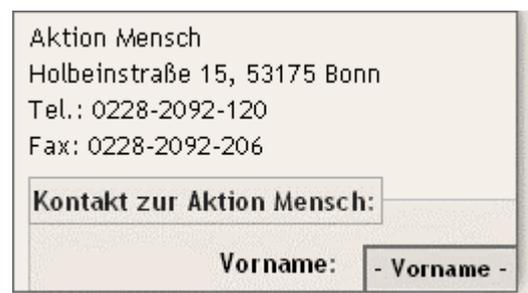
Wie wir an den vergangenen Tagen gesehen haben, werden nicht immer alle Hilfen zur Barrierefreiheit, die man in seine Seiten einbauen kann, auch tatsächlich von allen assistiven Technologien unterstützt. Die folgende Anforderung regelt das Verhalten einer Website für genau den Fall: ist das gesamte Angebot immer noch mit älteren oder nicht Standard-konformen assistiven Technologien verwendbar, wenn diese die eingesetzten neuern Techniken nicht beherrschen?

Anforderung 10	Die Verwendbarkeit von nicht mehr dem jeweils aktuellen Stand der Technik entsprechenden assistiven Technologien und Browsern ist sicherzustellen, so weit der hiermit verbundene Aufwand nicht unverhältnismäßig ist.
----------------	--

Sollte sich in der Entwicklung einer Website allerdings herausstellen, dass veraltete oder insbesondere nicht Standard-konforme Zugangsformen nur mit unverhältnismäßig hohen Mitteln oder auch gar nicht zu unterstützen sind, so können diese außen vor bleiben. Die BITV spielt damit indirekt den Ball wieder an die Hersteller zurück, die für eine Verbesserung Ihrer Produkte sorgen müssen.

Manchmal sind es aber auch die kleinen Dinge, über die Benutzer assistiver Technologien oder auch "ganz normale" Browser stolpern. Mit einer [Ausnahme](#) versteht bis heute kein Browser den kompletten Sprachumfang von HTML 4, obwohl dieser Standard mittlerweile schon [in die Jahre gekommen ist](#). So kann es unter bestimmten Umständen in manchen Textbrowsern oder assistiven Tools dazu kommen, dass Inhalte, die im Code eigentlich sauber voneinander getrennt sind, trotzdem ohne Trennung nacheinander ausgegeben werden.

Sie können sich nicht darauf verlassen, dass einige selten angewandte Tags, die laut HTML-Spezifikation blockbildend sind, auch als solche erkannt werden. Bei unseren Tests während des EfA-Relaunchs mussten wir zum Beispiel feststellen, dass manche Tags in assistiven Tools oder älteren Browsern gar nicht erkannt wurden und diese den Inhalt einfach zusammen mit dem Inhalt des nächsten (ihnen auch unbekanntem) Tags als eine lange Zeile ausgaben. Im nebenstehenden Screenshot ist dies eine Folge von `<address>`, `<div>`, `<fieldset>`, `<legend>`, `<label>` und `<input>`, was sich in etwa so anhörte:



Aktion Mensch
 Holbeinstraße 15, 53175 Bonn
 Tel.: 0228-2092-120
 Fax: 0228-2092-206

Kontakt zur Aktion Mensch:

Vorname:

Fax: 0228-2092-206 Kontakt zur Aktion Mensch Vorname: [Textfeld]

Das gleiche passierte auch am Ende des Formulars, hier wurde auf einer Braillezeile der "Absenden"-Button in einer Reihe mit dem darauf folgenden Text ausgegeben, obwohl sich mehrere Schlußtags von Blockelementen dazwischen befanden. Die Lösung bestand darin, einfach zusätzliche, redundante Zeilenschaltungen mit `
` einzufügen, um die Vorgabe Die Verwendbarkeit von nicht mehr dem jeweils aktuellen Stand der Technik entsprechenden assistiven Technologien und Browsern ist sicherzustellen zu erfüllen.

Bedingung 10.1	Prio. 1	Das Erscheinenlassen von Pop-Ups oder anderen Fenstern ist zu vermeiden. Die Nutzerin, der Nutzer ist über Wechsel der aktuellen Ansicht zu informieren.
----------------	---------	--

Was heißt das?

Hier beginnt der Bereich, in dem sich Usability und Accessibility überschneiden. Das unerwünschte Öffnen von Pop-Up-Fenstern ist generell unter Internetnutzern nicht sonderlich beliebt und wirft für alle Probleme in der Nutzbarkeit eines Webangebots auf.

Der zweite Teil der Bedingung bezieht sich übrigens nicht nur auf Pop-Ups, sondern auch auf Links, die selbsttätig, d.h. ohne explizite Anforderung durch den Nutzer in einem neuen Fenster geöffnet werden. Wenn Ihre Seiten sauberes XHTML verwenden, sollte dies eh nicht passieren, da das hierfür notwendige Attribut `target` ersatzlos gestrichen wurde. Wenn Sie sich trotzdem auf die Suche nach Links machen wollen, die sich in neuen Fenstern öffnen, sollten Sie sich das folgende Codeschnipsel in Ihr User Style Sheet legen:

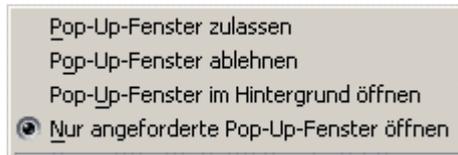
```
:link:hover[target="_blank" ] ,
:visited:hover[target="_blank" ] ,
:link:hover[target="_new" ] ,
:visited:hover[target="_new" ] {
color : white !important;
background : red !important;
}
```

Ab sofort wird Ihr Browser alle Links, die per `target="_new"` oder `target="_blank"` in einem neuen Fenster geöffnet werden sollen, leuchtend rot hinterlegen.

Für alle Nutzergruppen, gleich ob behindert oder nicht, bleibt bei sich automatisch öffnenden Fenstern das Problem, dass die Ansicht sich unaufgefordert ändert und der sog. Clickstream unterbrochen wird und sich nicht mehr über die History des Browsers (den "Zurück"-Knopf) wiederherstellen lässt.

Wie können Sie das testen?

Treue Leser kennen es schon: öffnen Sie Opera, gehen in die Schnelleinstellungen (F12) und deaktivieren Sie die Pop-Up-Fenster komplett über den Menüpunkt "Pop-Up-Fenster ablehnen". Wenn sich nun Ihre Website trotzdem uneingeschränkt nutzen können, haben Sie den Test bestanden.



Ein Beispiel:

Pop-Up-Fenster werden üblicherweise über JavaScript geöffnet, in dem bei einem Klick auf einen Hyperlink oder ein anderes Event (dies kann bereits das Laden der Seite sein, also `onLoad`) selbsttätig ein weiteres Fenster geöffnet wird. Diesem Fenster können dann durch das Skript bestimmte Parameter mitgegeben werden - Größe, Position, mit oder ohne Toolbars usw. Wenn der Entwickler sich hier nur auf die Fähigkeit zur Ausführung von JavaScript verlässt, öffnet sich für Benutzer ohne diese Skriptsprache gar nichts:

```
<a href="javascript:foo()">Hyperlink</a> ,
```

da im href-Attribut kein URL, sondern nur der Aufruf einer JavaScript-Funktion steht. Daher sollte in den Hyperlink immer als Fallback ein normaler HTML-Link eingebaut sein, der sich an Stelle dessen öffnet, sobald JavaScript nicht ausführbar ist. Der Aufruf der JavaScript-Funktion folgt danach, aber wie im folgenden Beispiel zusätzlich mit einem weiteren Event-Handler für den Fall, dass keine Maus benutzt wird:

```
<a href="foo.html"
  onclick="foo();
  onkeypress="foo();
  return false;">Hyperlink</a>
```

Bedingung 10.2	Prio. 1	Bei allen Formular-Kontrollelementen mit implizit zugeordneten Beschriftungen ist dafür Sorge zu tragen, dass die Beschriftungen korrekt positioniert sind.
Bedingung 12.4	Prio. 1	Beschriftungen sind genau ihren Kontrollelementen zuzuordnen.

Was heißt das?

Sie kennen das aus Ihrem Betriebssystem: bei Radiobuttons oder Checkboxes müssen Sie nicht unbedingt das Element selber anklicken, meistens reicht es, wenn Sie auf den danebenstehenden Text (das sog. Label) klicken und schon ist das Häkchen gesetzt. Genau das geht seit HTML 4 auch mit allen Elementen in Web-Formularen, nur wird es viel zu selten eingesetzt.

Wie können Sie das testen?

Diesmal dürfen Sie tatsächlich mit einer Maus testen und brauchen nichts in Ihrem Browser abzuschalten! Nehmen Sie sich einfach alle Formulare Ihrer Website nacheinander vor und testen Sie, ob die Kontrollelemente auch über den Klick auf den danebenstehenden Text aktivieren lassen. Wenn Radiobuttons (Entweder-Oder-Auswahlen) in einem Formular vorhanden sind, sollten es logischerweise immer mindestens zwei sein, Checkboxes können auch alleine auftreten.

Bei Textfeldern sollte die Einfügemarke nach dem Klick auf das Label im entsprechenden Textfeld stehen, so dass Sie sofort mit der Eingabe beginnen können. Auswahlboxen werden bei der Aktivierung meistens durch eine Umrandung hervorgehoben, die Auswahl selber lässt sich dann mit den Pfeiltasten (rauf/runter) vornehmen.



Achten Sie bei Ihren Tests auch darauf, dass durch den Klick das richtige, d.h. zu dem geklickten Label gehörende Kontrollelement aktiviert wird.

Ein Beispiel:

In der Gestaltung Ihrer Formulare sollten Sie sich von den Interface-Konventionen der großen Betriebssysteme leiten lassen, da diese den Benutzern höchstwahrscheinlich vertraut sind. So sollten die Labels von Checkboxes und Radiobuttons im Idealfall immer rechts neben dem jeweiligen Element stehen, bei Texteingabefeldern kann dies sowohl links daneben als auch unmittelbar darüber sein. Im Magazin von 'Einfach für alle' finden Sie [mehrere Tutorials](#), wie Sie Formulare barrierefrei und benutzerfreundlich gestalten und programmieren können.

```
<label for="email">Ihre <span lang="en">E-Mail</span>-Adresse: </label>
```

```
<input type="text" id="email"
tabindex="3" value="- E-Mail -,
title="Hier bitte Ihre E-Mail Adresse eingeben" />
```

Bedingung 10.4	Prio. 2	Leere Kontrollelemente in Eingabefeldern und Textbereichen sind mit Platzhalterzeichen zu versehen.
----------------	---------	---

Was heißt das?

Eine Vorgabe, die dazu gedacht ist, für Benutzer assistiver Tools den Umgang mit Textfeldern in Formularen zu vereinfachen. Entstanden ist dies aus der Überlegung, dass Textfelder hiermit besser als solche erkennbar werden und noch zusätzliche Informationen über Ihren Zweck mitliefern.

Leider hat sich diese Bedingung bei mehreren Tests mit Braillezeilen-Benutzern als eher hinderlich erwiesen. Die wenigsten Browser und auch nicht alle assistiven Tools markieren diesen vorbelegten Text, damit dieser bei einer Eingabe überschrieben werden kann. Home Page Reader, JAWS und WindowEyes beherrschen diese Technik, Webwizard und Lynx leider nicht. Also ist der Anwender u.U. darauf angewiesen, diesen Voreintrag zu erkennen und von Hand zu löschen, was leider in vielen Tests nicht geschehen ist.

Das Problem wurde dadurch noch verstärkt, dass es in unseren Testcases eine Plausibilitätsabfrage (server- und/oder clientseitig) gab, die z.B. die E-Mail-Adresse auf bestimmte Regeln überprüfte, um Fehleingaben zu vermeiden. In den Tests bekamen die Testpersonen dann immer die Fehlermeldung zurück:

```
"user@domain.de - Ihre E-Mail Adresse - , ist keine gültige E-Mail Adresse" .
```

Wie können Sie das testen?

Für diesen Test brauchen Sie natürlich keinen speziellen Browser, die einzige Bedingung an das Ausgabegerät ist die Fähigkeit, Formulare zu verarbeiten. Von da an sind Sie allerdings auf sich selbst gestellt, hier hilft nur die manuelle Kontrolle aller Formulare Ihrer Website.

Ein Beispiel:

Wie Sie gesehen haben, werden Formulare durch die Einhaltung dieser Bedingung unter Umständen fehleranfälliger und unbequemer zu bedienen. Beim Relaunch von 'Einfach für Alle' stellte sich natürlich dasselbe Problem. Die Lösung hierfür war, die Textfelder zwar vorzubelegen, diesen Eintrag aber per JavaScript zu löschen, sobald das Textfeld aktiviert wird:

```
<form method="get" action="/suche/search.pl">
  <fieldset>
    <label for="terms" accesskey="5">nach:</label><br />
    <input type="text" id="terms" value="eingeben"
      title="Suchbegriff hier eingeben [Accesskey 5]"
      onblur="if(this.value=="')this.value='eingeben';"
      onfocus="if(this.value=='eingeben')this.value="';" /><br />
    <input type="submit" id="find" value="Finden" />
  </fieldset>
</form>
```

Dieses bringt zumindest bei Browsern mit JavaScript-Funktionalität den Komfort, den Eintrag nicht von Hand löschen zu müssen, hilft aber bei abgeschaltetem JavaScript auch nicht weiter.

Für die anstehende Überarbeitung der BITV wäre es sicher sinnvoll, diese Vorgabe auf Textfelder einzuschränken, die üblicherweise nicht mit einem Label ausgestattet werden, also z.B. die Eingabefelder von Suchabfragen. Dies vor allem vor dem Hintergrund dass die zu einem Textfeld gehörigen Labels bereits über `tabindex` und/oder `accesskey` angesprochen werden und sich damit der Sinn des Textfeldes bereits erschließt.

Bedingung 10.5	Prio. 2	Nebeneinanderliegende Hyperlinks sind durch von Leerzeichen umgebene, druckbare Zeichen zu trennen.
----------------	---------	---

Was heißt das?

Mit dieser Bedingung soll verhindert werden, dass Aufreihungen von Links ohne Trennung nacheinander wiedergegeben werden und so nicht mehr differenzierbar sind. Dies gilt insbesondere für Sprachausgaben, die zudem die unangenehme Eigenschaft haben, bei langen Wortfolgen ohne Interpunktion oder sonstige Trennzeichen in der Satzmelodie abzufallen, was den Lese- oder besser Hörfluss nicht gerade vereinfacht. Zudem wird die Bedienung nebeneinander stehender Links für Menschen mit motorischen Behinderungen einfacher, wenn diese etwas voneinander entfernt sind.

Nebeneinander heißt nicht nur, dass diese Zeichen in der grafischen Darstellung horizontal nebeneinander liegen. So können Zeichen durch CSS-Positionierung an verschiedenen Stellen im Dokument auftauchen, im Quelltext jedoch unmittelbar aufeinander folgen. In einer nicht-grafischen Darstellung würden diese dann ohne eine adäquate Trennung wieder nebeneinander stehen.

Druckbar heißt, dass dies Zeichen aus dem verwendeten Zeichensatz sein müssen, die keine Leerzeichen (` `) sind. Auch Bilder als alleinige Trennung von aufeinanderfolgenden Links sind nicht zulässig. Bei nebeneinanderstehenden Links hat sich mittlerweile das Pipe-Zeichen (`|`), umgeben von jeweils einem Leerschritt eingebürgert.

Wie können Sie das testen?

Dieser Test ist am einfachsten mit einem Werkzeug wie Bobby durchzuführen, da dieser Ihre Seiten auch nach diesem Fehler untersucht. Wenn Sie Ihre Seiten von [Bobby](#) nach den Web Content Accessibility Guidelines 1.0 überprüfen lassen, wird dieser und die entsprechenden Meldungen zurückgeben.

Ein Beispiel:

Ein Beispiel für diese Technik finden Sie am Ende jeder Seite bei 'Einfach für alle'. Zusätzlich zu den Pipe-Zeichen wird für standard-konforme Browser noch ein kleines Icon zur besseren Differenzierung der Links eingefügt:



Tag 11: Standards

Die Verwendung von offenen Standards garantiert nicht nur die Interoperabilität Ihres Internetangebots, sondern spart bei Wartung und Pflege bares Geld:

Anforderung 11	Die zur Erstellung des Internetangebots verwendeten Technologien sollen öffentlich zugänglich und vollständig dokumentiert sein, wie z.B. die vom World Wide Web Consortium entwickelten Technologien.
----------------	--

Was heißt das?

Man beachte die Einschränkung in dieser Vorgabe (**wie z.B.** die vom World Wide Web Consortium entwickelten Technologien), in der die W3C-Empfehlungen nur als *eine mögliche* Technologiebasis genannt werden. Einer der Gründe für diese Einschränkung ist die Tatsache, dass das W3C keine normative Kraft wie das [DIN](#) oder die [ISO](#) besitzt, sondern lediglich eine privatwirtschaftlich organisierte Vereinigung ist. Daher wäre es auch falsch, bei den Empfehlungen des W3C von *Standards* zu sprechen, da diese Empfehlungen ([Recommendations](#)) nur eines sind: freundliche Empfehlungen.

Aus dieser Anforderung folgt unter anderem, daß Sie bei Ihren Seiten auf proprietäres, d.h. hersteller-spezifisches Markup grundsätzlich verzichten sollten. Was dies für die Webentwicklung bedeutet, haben wir bereits am [Tag 3 dieser Serie](#) erläutert.

Mit dieser "Öffnungsklausel" schließt die BITV auch andere de-Facto-Standards ein, die nicht vom W3C kontrolliert werden: JavaScript ist als [ECMA-262](#) vollständig dokumentiert und normiert worden. Auch das Flash-Format ist vom Hersteller Macromedia vor einiger Zeit offengelegt und dokumentiert worden, so dass es hier wohl ebenfalls eingeschlossen ist.

Allerdings ist es für eine Verordnung aus formaljuristischen Gründen nicht möglich, sich direkt auf Standards zu beziehen, die nicht der regierungsamtlichen Kontrolle unterliegen. Daher diese etwas vagen Formulierungen, die uns auch noch an anderen Stellen der Verordnung begegnen.

Bedingung 11.1	Prio. 1	Es sind öffentlich zugängliche und vollständig dokumentierte Technologien in ihrer jeweils aktuellen Version zu verwenden, soweit dies für die Erfüllung der angestrebten Aufgabe angemessen ist.
Bedingung 11.2	Prio. 1	Die Verwendung von Funktionen, die durch die Herausgabe neuer Versionen überholt sind, ist zu vermeiden.

Was heißt das?

Älteren Auszeichnungssprachen wie HTML 3.2, aber strenggenommen auch HTML 4.x ist damit die Geschäftsgrundlage entzogen, da es mit XHTML bereits eine neuere Version gibt. Diese Bedingung gilt insbesondere für Sprachelemente älterer HTML-Versionen wie `<font` und ähnliche Tags, die in [HTML 4](#) als deprecated gekennzeichnet sind.

Wie können Sie das testen?

In manchen Browsern wie dem iCab für Apple Macintosh lässt sich die Darstellung nicht-standardkonformer Tags gänzlich unterbinden, aber meistens reicht es schon, einen modernen Browser

wie Opera 7 oder Mozilla 1.3 zum Testen heranzuziehen. Diese Browser verstehen zwar unter Umständen auch nicht alle aktuellen HTML–Tags oder haben noch Fehler in der Darstellung komplexer Seiten. Sie zeichnen sich aber vor allem dadurch aus, daß sie komplett auf die Unterstützung von proprietärem Markup verzichten.

Bei den Tests zu dieser Vorgabe helfen wieder Online–Werkzeuge wie der [W3C–Validator](#): Viele Hilfen zur Barrierefreiheit wurden erst mit der Version 4 in die Seitenbeschreibungssprache HTML eingeführt. Wenn der Validator bei der Überprüfung Ihrer Seiten zu dem Ergebnis kommt, daß es sich um HTML 3.2 oder noch Älteres handelt, ist leider einiges an Nacharbeit nötig. Selbst wenn die Seiten formal nach diesen veralteten de–fakto–Standards validieren, sind sie damit noch nicht automatisch auf der technischen Ebene barrierefrei – eher das Gegenteil ist wahrscheinlich der Fall.

Ein Beispiel:

Die Verwendung der W3C–Vorgaben bringt noch einen nicht zu unterschätzenden Vorteil: nicht nur dass Ihre Seiten insgesamt schlanker und damit performanter werden, auch die nachträgliche Pflege wird stark vereinfacht. Ein mit der Pflege einer Website beauftragter Entwickler braucht sich nicht mehr in den individuellen Code des ursprünglichen Programmierers einzuarbeiten, sondern kann sein Wissen der standard–konformen Entwicklung sofort produktiv einsetzen.

Wenn man mit Entwicklern spricht, die den Schritt zu sauberem Code (vorzugsweise XHTML und CSS) und den gängigen Maßnahmen zur Barrierefreiheit vollzogen haben, hört man immer wieder die selben Antworten: Der Lernaufwand ist zwar zunächst vielleicht hoch, vor allem, weil man einige Gewohnheiten ablegen und mühsam erworbenes Wissen um verschachtelte Tabellenlayouts "entlernen" muß. Aber trotzdem bereut kaum jemand diesen Schritt, da er ein wesentlich einfacheres, strukturierteres Arbeiten ermöglicht.

Bedingung 11.3	Prio. 1	Soweit auch nach bestem Bemühen die Erstellung eines barrierefreien Internetangebots nicht möglich ist, ist ein alternatives, barrierefreies Angebot zur Verfügung zu stellen, das äquivalente Funktionalitäten und Informationen gleicher Aktualität enthält, soweit es die technischen Möglichkeiten zulassen. Bei Verwendung nicht barrierefreier Technologien sind diese zu ersetzen, sobald aufgrund der technologischen Entwicklung äquivalente, zugängliche Lösungen verfügbar und einsetzbar sind.
----------------	---------	--

Was heißt das?

Hier stellt sich natürlich die Frage, warum man überhaupt ein Parallel–Angebot mit äquivalenten Funktionalitäten und Informationen gleicher Aktualität bereitstellen sollte. Wenn letzteres die identische Funktionalität in barrierefreier Form zur Verfügung stellt, wäre es dann nicht sinnvoller, gleich auf die Barrieren zu verzichten?

Bei oberflächlicher Betrachtung stimmt dies sicher auch für die meisten Webangebote. Die BITV zeigt durch diese Vorgabe aber einen Königsweg auf, um als Anbieter auf die sehr speziellen Bedürfnisse mancher Benutzergruppen reagieren zu können. So können Sie nicht davon ausgehen, dass [gehörlose Menschen](#) oder Menschen mit Lernbehinderungen über die selbe Kompetenz im Umgang mit Schriftsprache verfügen wie andere Nutzergruppen. Hier ist oftmals der Umweg über multimediale Formate nötig, um Inhalte in optimaler Form transportieren zu können.

Tipp:

Begehen Sie allerdings bitte nicht den Fehler, bestimmte Nutzergruppen in ein separates Angebot zu

nötigen. Viele hörbehinderte Menschen haben nur geringe bis gar keine Probleme mit Schriftsprache – diese ist aber trotzdem für sie nur die 'erste Fremdsprache'. Daher sollte ein alternatives Angebot immer nur optional sein.

Der Zweite Satz der Bedingung kümmert sich um die technische Aktualität solcher multimedialen Angebote. Inhalte, die mit älteren Versionen von Flash oder Director erstellt wurden, waren für assistive Werkzeuge wie Sprachausgaben oder Eingabegeräte jenseits der Maus nicht benutzbar. Die neuen Versionen der Entwicklungsumgebungen Flash MX und Director MX unterstützt aber mittlerweile die MSAA-Schnittstelle, über die aktuelle Versionen von Screenreadern wie JAWS an die Inhalte eines Flash-Films gelangen können. Andere Hersteller assistiver Werkzeuge werden diesem Beispiel sicher folgen, so daß Sie ab einem bestimmten Grad der Verbreitung ohne schlechtes Gewissen solche multimedialen Formate anbieten werden können.

Aber auch wenn sich mit diesen Werkzeugen mittlerweile Resultate erzielen lassen, die nach den gängigen Vorgaben barrierefrei sind – der Aufwand für zusätzlich eingebaute Untertitel (closed captioning) ist nach wie vor beträchtlich. Bis die Bedingung soweit es die technischen Möglichkeiten zulassen wirklich durchsetzbar ist, wird wohl noch einige Zeit verstreichen.

Ein Beispiel:

Ein Beispiel für multimediale Inhalte, die für bestimmte Nutzergruppen absolut barrierefrei sind, zeigt die Aktion Mensch mit der Site zur Ausstellung "[Der \(im-\)perfekte Mensch](#)". Hier wurden zum einen die schwierigen Texte in die sogenannte "Leichte Sprache" übersetzt, um Menschen mit Lernbehinderungen oder Leseschwäche, aber auch nicht-Muttersprachlern oder Grundschulern den inhaltlichen Zugang zu ermöglichen.



Durch die Programmierung kann der Besucher beliebig zwischen "Leichter Sprache" und Schriftsprache hin- und zurück schalten, oder auch den gesamten Rundgang durch die Ausstellung nur in "Leichter Sprache" erleben.

Zusätzlich wird die gesamte Ausstellung noch multimedial als virtueller Rundgang mit QuickTime VR präsentiert, so daß die Inhalte auch ohne Kenntnis der Sprache "erlebt" werden können. Die Hinführung zum Text erfolgt in den Filmen wiederum durch anklickbare Ausstellungsobjekte, für die auf Wunsch der entsprechende Katalogtext angezeigt wird.



Tag 12: Orientierung

Vielen Benutzern assistiver Technologien wird der Zugang bereits durch die mangelnde Übersichtlichkeit einer Website verwehrt. Absicht steckt aber in den seltensten Fällen dahinter, sondern meistens das mangelnde Wissen um die Funktionsweise der assistiven Werkzeuge behinderter Menschen.

Anforderung 12	Der Nutzerin, dem Nutzer sind Informationen zum Kontext und zur Orientierung bereitzustellen.
-----------------------	--

Ohne das Wissen um die eigentümliche Funktionsweise z.B. eines Screenreaders liest sich diese Anforderung zunächst erst einmal wie eine Selbstverständlichkeit. Jede Website stellt doch über die Navigation Informationen zur Orientierung bereit. Dass durch das bloße Vorhandensein von eingängigen und nachvollziehbaren Navigationsmechanismen alleine nicht schon eine Barrierefreiheit sichergestellt ist, wird durch die folgenden Bedingungen verdeutlicht:

Bedingung 12.1	Prio. 1	Jeder Frame ist mit einem Titel zu versehen, um Navigation und Identifikation zu ermöglichen.
Bedingung 12.2	Prio. 1	Der Zweck von Frames und ihre Beziehung zueinander ist zu beschreiben, soweit dies nicht aus den verwendeten Titeln ersichtlich ist.

Was heißt das?

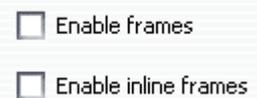
Für die Verwendung der Rahmentechnik (sog. Frames) kann es trotz [vieler Vorbehalte](#) in einzelnen Fällen durchaus gute Gründe geben. Allerdings gilt es dabei einige Dinge zu beachten, die sich aus der Funktionsweise nicht-grafischer Zugangsarten ergeben.

Wenn Sie [Frames benutzen](#) sollten Sie zum einen das `<noframes>`-Tag einsetzen, um entweder direkt die Inhalte darzustellen oder um zumindest auf eine framelose Variante zu verweisen.

Zum anderen sollten die Frames auch sinnbildend benannt sein, da viele nicht-grafische Browser die Benennung der Frames auswerten und darstellen. Wenn Ihre Frames in einem solchen Browser als `name="frame_links_01"` und `name="frame_rechts_02"` erscheinen, ist für den Anwender nicht mehr nachvollziehbar, was sich hinter diesen Namen verbirgt. Daher sollten Navigationsframes beispielsweise mit `name="navigation"` und Inhaltsframes mit `name="inhalt"` gekennzeichnet sein.

Wie können Sie das testen?

Mit grafischen Browsern ist dieser Test einigermaßen schwierig, da diese seit mehreren Generationen Frames verstehen und diese nicht abschaltbar sind. Eine der wenigen Ausnahmen ist Opera: hier finden Sie in den Voreinstellungen des Browsers ein Kontrollkästchen zur Deaktivierung von Frames. Wenn dieses nicht angekreuzt ist, stellt Opera den Inhalt des `<noframes>`-Tags dar (sofern vorhanden).



Wenn Sie Ihre Frame-Seiten auch mit assistiven Werkzeugen testen möchten, der kostenlose [WebFormator](#) der Firma Frank Audiodata oder auch der [Home Page Reader](#) zeigen die Struktur eines

Framesets inklusive der (hoffentlich korrekten) Benennungen der Frames und der in den einzelnen Frames vorgefundenen Textinhalte an.

Ein Beispiel:

Die [Website des Lufthansa City Center](#) benutzt Frames für das visuelle Layout der Seiten. Im folgenden Screenshot sehen Sie die Darstellung in einem grafischen Browser, wo bereits bei einer Auflösung von 640x480 Pixel wesentliche Inhalte im obersten Frame abgeschnitten und damit nicht mehr erreichbar sind.



Betrachtet man diese Seiten nun in einem nicht-grafischen Medium wie dem Textbrowser Lynx, so ist der folgende Text die einzig verwertbare Information, an die der Anwender gelangt:

```
FRAME: head
FRAME: body
FRAME: foot
```

Diese website benutzt Frames. Sie können sich bei [Microsoft](#) kostenlos einen Browser herunterladen, der Frames abbilden kann.

Die Firma Microsoft ist zwar Marktführer bei den grafischen Desktop-Browsern, stellt aber noch lange nicht für jedes Betriebssystem einen Browser her.

Bedingung 12.3	Prio. 1	Große Informationsblöcke sind mittels Elementen der verwendeten Markup-Sprache in leichter handhabbare Gruppen zu unterteilen.
----------------	---------	--

Was heißt das?

Bei der Gestaltung mit CSS geschieht diese Unterteilung in handhabbare Gruppen schon fast zwangsläufig. Diese Sprache entfaltet ihre ganze Eleganz erst, wenn der Code nach einem festen Regelwerk strukturiert ist. Diese Regeln werden von externen Style Sheets benötigt, um eine Seite überhaupt formatieren zu können. Die hierbei zwangsläufig anfallenden Informationsblöcke (Inhaltsbereiche, Navigationsleisten etc.) sind bei entsprechender Verwendung von strukturierenden Elementen auch mit zusätzlichen Navigationshilfen ausrüstbar, um die sich die nächste Bedingung kümmert.

Bedingung 13.6	Prio. 2	Inhaltlich verwandte oder zusammenhängende Hyperlinks sind zu gruppieren. Die Gruppen sind eindeutig zu benennen und müssen einen Mechanismus enthalten, der das Umgehen der Gruppe ermöglicht.
----------------	---------	---

Was heißt das?

Durch diese Bedingung sollen alternative Ausgabemedien in die Lage versetzt werden, den Benutzer bei der Navigation in nicht-grafischen Umgebungen zu unterstützen. In der Darstellung auf mobilen Endgeräten, aber auch in der Sprachausgabe fehlt der visuelle Kontext, mit denen sich ein Benutzer von

grafischer Browsern auf einer Seite orientieren und ohne großen Aufwand zwischen den verschiedenen Inhaltsblöcken einer Seite springen kann.

Zur Unterstützung dieser Nutzer ist es sinnvoll, z.B. Navigationsblöcke mit einem Verweis innerhalb der Seite überspringbar zu machen, so dass diese nicht immer wieder vorgelesen werden müssen, bevor man an den eigentlichen Inhalt der Seite gelangt.

Dies geschieht in HTML über lokale Anker (``), die auf einen Punkt einer Seite mit einer bestimmten ID (`<div id="inhalt">`) verweisen. Dies sind genau die ID's, die man zur Identifizierung von Blöcken einsetzt, um per CSS Stile zuweisen zu können. Der leichte Einbau einer solchen Navigation innerhalb einer Seite ist also fast schon ein Abfallprodukt der CSS-Formatierung und entsprechend einfach umzusetzen. Daher sollten Sie diese Vorgabe abweichend von der BITV durchaus mit Priorität 1 behandeln.

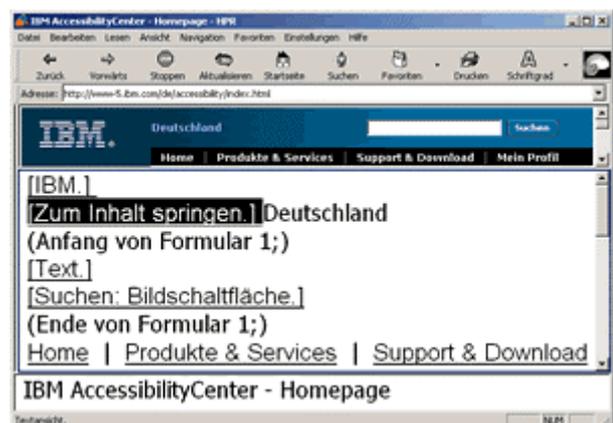
Wie können Sie das testen?

Wenn diese lokalen Links im Design Ihrer Seite sichtbar angelegt sind, reicht hier natürlich ein normaler grafischer Browser zur Überprüfung. Mittlerweile ist es jedoch gängige und auch durchaus legitime Praxis, diese Links per CSS für grafische Browser zu verstecken, so dass diese nur in alternativen Medien ausgegeben werden. Um diesen Konstrukt zu überprüfen müssen sie also in Ihrem Browser CSS abschalten. Am einfachsten geht dies in so gut wie allen Browsern mit dem bereits an [Tag 6](#) vorgestellten [ToggleCSS-Favelet](#).

Alternativ können Sie natürlich auch einen Sprachbrowser wie den IBM Home Page Reader oder einen Screenreader verwenden und Ihren Monitor ausschalten. Versuchen Sie dann einmal, sich nur per Sprachausgabe und Tastaturbedienung auf Ihren Seiten zu orientieren. Gelingt Ihnen dies in einer angemessenen Zeit (und ohne den Monitor wieder anzuschalten!), so haben Sie den Test bestanden.

Ein Beispiel:

Das [IBM Accessibility Center](#) nutzt genau diese Möglichkeit der versteckten Navigation, damit der Besucher unmittelbar zum Inhalt gelangen kann. In der grafischen Darstellung befindet sich am Beginn der Seite eine Navigationsleiste mit dem Logo und Verweisen in die Unterbereiche der Site. Im Code wiederum befindet sich unmittelbar nach dem Logo ein lokaler Link auf den Inhalt der Seite [Zum Inhalt springen.], wie im folgenden Screenshot aus dem Home Page Reader zu sehen ist:



Tag 13: Gebrauchstauglichkeit

Klassische Usability-Faktoren haben einen ebenso starken Einfluss auf die Barrierefreiheit eines Angebots wie die bislang besprochenen technische Maßnahmen. Während diese rein technischen Merkmale aber weitestgehend maschinell, d.h. mit Validatoren und anderen Prüfwerkzeugen verifizierbar sind, lassen die heutige und morgige Vorgabe einigen Spielraum für Interpretationen.

Anforderung 13	Navigationsmechanismen sind übersichtlich und schlüssig zu gestalten.	
Bedingung 13.1	Prio. 1	Das Ziel jedes Hyperlinks muss auf eindeutige Weise identifizierbar sein.

Was heißt das?

Der Text eines Hyperlinks, sprich der gesamte verlinkte Text muss auch ohne seinen Satzzusammenhang noch zuverlässige Aussagen über das Ziel des Links machen. Viele assistive Browser, aber auch immer mehr herkömmliche grafische Browser können als Orientierungshilfe die Links einer Seite zusammenfassen und getrennt darstellen. Wenn Ihre Links also nicht eindeutig benannt sind, sind sie für diese Hilfen nicht brauchbar. Zudem ist der Text eines Links auch genau der Text, der in den meisten Browsern für ein Lesezeichen (Favorit oder Bookmark) benutzt wird, wenn ein Besucher sich ein solches Bookmark für einen Link setzt.

Tests mit Netz-Neulingen haben gezeigt, dass diese oftmals verwirrt sind, wenn ein Link sie auf eine externe Seite führt, die von einem anderen Server kommt und damit zwangsläufig ein anderes Design präsentiert. Daher sollten Sie externe Links als solche ankündigen und entsprechend kennzeichnen. Gängige Methoden sind hierfür kleine GIFs, die einem externen Link vorgeschaltet sind oder auch das `title`-Attribut des Link-Ankers, in das Sie zusätzliche Informationen einfügen können.

Wie können Sie das testen?

Ihre Seiten nun manuell nach schlüssigen Linktexten zu durchforsten wäre ein viel zu hoher Aufwand. Daher sollten Sie sich das folgende Bookmarklet von accessify.com in Ihren Internet Explorer installieren:

Links zusammenfassen

Ziehen Sie diesen Link hierzu einfach mit gedrückter Maustaste in die Favoritenleiste des Browsers. Wie bei den Bookmarklets der vergangenen Tage bekommen sie eine neue Schaltfläche mit der Aufschrift dieses Links. Durch Klick auf diese Schaltfläche wird eine neue Seite mit einer Zusammenfassung aller Links der aktuellen Webseite geöffnet. Bitte beachten Sie, dass hierfür JavaScript in Ihrem Browser aktiviert sein muss.



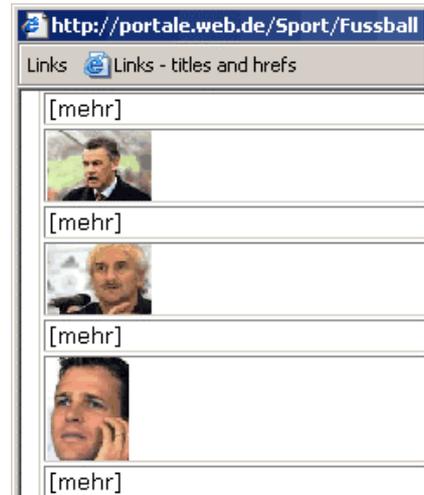
Inhalt	Inhaltsverzeichnis mit allen Artikeln nach Kategorien sortiert	http://www.einfachfueralle.de/artikel/bitvfueralle/tag12/
Blog	Tägliche News zur Barrierefreiheit	http://www.einfachfueralle.de/artikel/bitvfueralle/tag12/
Hilfe	Hilfe zur Bedienung und Orientierung	http://www.einfachfueralle.de/artikel/bitvfueralle/tag12/
vieler Vorbehalte	Artikel: Damit Sie nicht aus dem Rahmen fallen	http://www.einfachfueralle.de/artikel/bitvfueralle/tag12/
Frames benutzen	Externer Link: Suchabfrage bei Google nach 'Diese Website benutzt Frames'	http://www.google.com/search?q=Diese+Website+benutzt+Frames&btnG=Google

Ein Beispiel:

Viele News-Portale benutzen auf ihren Übersichtsseiten die Methode, Artikel mit sogenannten "Teasern" kurz anzureißen und dann mit einem Link [mehr . . .] auf den vollen Text zu verweisen. Wenn dieser Link die einzige Information über das Linkziel bleibt, ist für Nutzer alternativer Ausgabeformen ohne den visuellen Kontext des Teasers nicht mehr erkenntlich, wohin dieser Link eigentlich führt. Eine Sprachausgabe würde hier unter Umständen also lediglich eine lange Reihe von

"Link: [mehr...] Link: [mehr...] Link: [mehr...]
Link: [mehr...]" vorlesen.

Einige Anbieter versuchen diese Problematik zu umgehen, indem statt eines mehr...-Links die Überschrift des Teasers verlinkt wird. In nicht-visuellen Umgebungen ist dies aber auch problematisch: man hört zunächst die Überschrift mit dem Link, und kann unter Umständen erst nach dem darauffolgenden Teaser entscheiden, ob man diesen Link weiter verfolgen möchte. Hierzu muss der Nutzer dann erst wieder umständlich zur Überschrift zurück navigieren.



Bedingung 13.2	Prio. 1	Es sind Metadaten bereitzustellen, um semantische Informationen zu Internetangeboten hinzuzufügen.
Bedingung 13.9	Prio. 2	Soweit inhaltlich zusammenhängende Dokumente getrennt angeboten werden, sind Zusammenstellungen dieser Dokumente bereitzustellen.

Was heißt das?

Semantik bezeichnet in der Sprachwissenschaft eigentlich die Lehre von der Bedeutung; dieser Begriff wird aber im Web mittlerweile etwas liberaler eingesetzt, um beschreibende Informationen eines Webangebots zu definieren.

Zu diesen beschreibenden Metadaten gehören zum Beispiel Informationen zum Kontext der jeweiligen Seite. Wenn ein Dokument eine Folge aus einer Serie ist (wie dieses Dokument hier), so sollten Sie die Hierarchie dieser Serie mit sogenannten Link Relations abbilden. Die wichtigsten Tags hierfür sind `<link rel="start">`, `<link rel="previous">`, `<link rel="next">` und `<link rel="last">`.

Diese Metadaten können aber auch Verweise auf ein Glossar, eine Richtlinie zum Datenschutz (privacy policy), ein Inhaltsverzeichnis, eine Suchfunktion oder eine Hilfe-Seite sein.

Wie können Sie das testen?

Aktuelle Browser wie Opera, Mozilla und iCab stellen diese Link Relations ohne weiteres Zutun in einer Menüleiste dar. Auch der Textbrowser Lynx wertet seit jeher diese Informationen aus und stellt diese am Beginn der Dokumente als Links dar. Da diese Menüleiste in den meisten Browsern abschaltbar ist, müssen Sie für diesen Test lediglich darauf achten, dass diese nicht versehentlich deaktiviert ist. Im Mozilla finden Sie den entsprechenden Befehl zur Anzeige im Menü "View / Show/Hide / Site Navigation Bar", in Opera 7 ist dies "View / Navigation Bar"

Ein Beispiel:

‘Einfach für Alle’ benutzt diese Meta-Informationen natürlich auch. Hierüber sind z.B. die Site-wide Suche, das Inhaltsverzeichnis oder auch ein Glossar erreichbar. Der folgende Screenshot zeigt die Umsetzung dieser Informationen im Browser Mozilla:



Bedingung 13.3	Prio. 1	Es sind Informationen zur allgemeinen Anordnung und Konzeption eines Internetangebots, z.B. mittels eines Inhaltsverzeichnisses oder einer Sitemap, bereitzustellen.
----------------	---------	--

Was heißt das?

Sie können sich nicht darauf verlassen, dass alle Besucher das Konzept hinter [Hypertexten](#) kennen, und sich über Links im Text eine komplette Website erschließen können, und sei diese interne Verlinkung auch noch so ausgefeilt. Viele Besucher erwarten auch im Web ein aus den Printmedien vertrautes Inhaltsverzeichnis zur schnellen Orientierung.

Begehen Sie aber nicht den Fehler vieler Anbieter und nennen Ihr Inhaltsverzeichnis wie hier in der BITV "Sitemap". Tests haben gezeigt, dass die Klickraten auf einen Link mit der Beschriftung "Sitemap" wesentlich geringer sind als auf den Link "Inhaltsverzeichnis" oder "Inhalt".

Bedingung 13.4	Prio. 1	Navigationsmechanismen müssen schlüssig und nachvollziehbar eingesetzt werden.
Bedingung 13.5	Prio. 2	Es sind Navigationsleisten bereitzustellen, um den verwendeten Navigationsmechanismus hervorzuheben und einen Zugriff darauf zu ermöglichen.

Was heißt das?

Die Einhaltung dieser Vorgabe hängt natürlich von vielen Faktoren ab, die über den Einflussbereich der BITV hinausgehen. Hier die verschiedenen Ansätze einer nachvollziehbaren und selbsterklärenden Navigation zu diskutieren, würde den Rahmen dieser Serie sprengen.

Wie können Sie das testen?

Dies ist eine Vorgabe, die Sie wirklich nur mit tatsächlichen Nutzern testen können. Ob Sie damit ein Usability-Labor beauftragen, oder ob Sie informell ein paar Freunde und Bekannte um einen Test bitten hängt vom zur Verfügung stehenden Budget und der Größenordnung des Projektes ab. Wichtig ist nur, dass diese Tests in einer möglichst frühen Phase der Entwicklung stattfinden. Nur so können die Erkenntnisse noch bis zum Start der Seiten effektiv und kostensparend umgesetzt werden.

Ein Beispiel:

Auf vielen Seiten staatlicher, aber auch privater Anbieter wird der Besucher von einer Navigation begrüßt, die nur für Eingeweihte nachvollziehbar ist. Ein häufiger Fehler ist es, die eigene Organisationsstruktur

des Unternehmens oder der Behörde als Grundlage für die Struktur der Website und der Navigation zu nehmen.

So erwartet das [Eisenbahn-Bundesamt](#) von seinen Besuchern, sich zum erfolgreichen Navigieren mit dem Organigramm der Behörde zu beschäftigen. Die wenigsten Benutzer werden jedoch wissen, was sich hinter Abteilung 1, Abteilung 2, Abteilung 3, Abteilung 4 und deren angeschlossenen Referaten verbirgt.

Für den Benutzer bedeutet dies, dass man sich auf jeder angesurften Website zunächst einmal mit der inneren Struktur des Anbieters beschäftigen muss, bevor man an die Inhalte gelangen darf – ein Aufwand, der sicher auch nicht lernbehinderten Menschen zuviel ist.



Bedingung 13.7	Prio. 2	Soweit Suchfunktionen angeboten werden, sind der Nutzerin, dem Nutzer verschiedene Arten der Suche bereitzustellen.
----------------	---------	---

Was heißt das?

Sie können nicht davon ausgehen, dass Ihre Besucher Boolesche Operatoren beherrschen, geschweige denn von deren Existenz wissen. Das leider immer noch häufig anzutreffende Schild Fahrradfahren und Ballspielen verboten ist ein Beleg hierfür: für Mathematiker oder Informatiker bedeutet dies, dass man während des Fahrradfahrens nicht auch noch gleichzeitig Ballspielen darf, das Eine *ohne* das Andere ist aber nach wie vor erlaubt.

Daher sollten Sie für Ihre Besucher verschiedene Schwierigkeitsgrade der Suche anbieten, zum Beispiel einmal eine simple Volltextsuche und zusätzlich eine Komfortsuche mit erweiterten Optionen und der Möglichkeit, logische Verknüpfungen zu benutzen. Immer wichtiger wird auch die Fehlertoleranz bei der Suche. So sind viele Such-Skripte mittlerweile in der Lage, Rechtschreibfehler in der Eingabe zu erkennen und ähnlich geschriebene oder phonetisch verwandte Suchtreffer mit anzuzeigen.

Tag 14: Verständnis

Am letzten Tag unserer Serie zur BITV geht es nochmals um die Barrierefreiheit auf der inhaltlichen Ebene. Selbst wenn eine Website alle Tests auf der technischen Ebene mit fliegenden Fahnen besteht, der Code validiert und die gängigen Prüfprogramme auf keine groben Schnitzer hinweisen, ist diese Website damit noch lange nicht für alle Besucher tatsächlich auch benutzbar.

Anforderung 14	Das allgemeine Verständnis der angebotenen Inhalte ist durch angemessene Maßnahmen zu fördern.	
Bedingung 14.1	Prio. 1	Für jegliche Inhalte ist die klarste und einfachste Sprache zu verwenden, die angemessen ist.

Was heißt das?

Kein Anbieter kann davon ausgehen, dass die Besucher seiner Website die in der Organisation des Anbieters gängigen Fachtermini beherrschen. Viele Texte, die sich an ein großes Publikum richten, werden trotzdem in einer Sprache verfasst, die fast schon ein abgeschlossenes Studium im jeweiligen Fachbereich voraussetzt. Sofern es sich nicht um Fachtexte handelt, sollten Sie also weitestgehend auf die beliebten Anglizismen und andere fremdsprachliche Formulierungen verzichten.

Fachsprache ist natürlich in vielen Bereichen durchaus gerechtfertigt. Nur sollten Inhalte, die für die Erledigung von Bedürfnissen des täglichen Lebens gedacht sind, auch so getextet sein, dass sie für die Benutzer keine unnötigen Hürden aufbauen. Dazu gehören nicht nur Behördengänge, sondern auch der Informationsbedarf der Internet-Nutzer oder der große Bereich des e-Commerce.

Viele Vorgaben und Ideen, die sich in den [Richtlinien zur Erstellung von Texten in leichter Sprache](#) für Menschen mit Lernbehinderungen finden, lassen sich auf auch Texte anwenden, die sich an ein großes Publikum mit unterschiedlichsten Sprachkompetenzen richten:

- *Verwendung von einfacher und unkomplizierter Sprache*
- *Keine abstrakten Begriffe*
- *Kurze Worte aus der Alltagssprache*
- *Persönliche Ansprache*
- *Praktische Beispiele / Bilder*
- *Kurze Sätze*
- *Keine Abkürzungen, Fremdwörter, Initialen*
- *Gleiche Begriffe für eine Sache*
- *Positive Sprache*

Diese Regeln sind der Mitschrift eines Beitrags von Cordula Edler zum Kongress [‘Mehr Wert für Alle’](#) in Trier entnommen.

Wie können Sie das testen?

Dies ist wieder eine der Vorgaben, die nicht maschinell zu überprüfen sind. Hier hilft tatsächlich nur, die Texte einer Website von potentiellen Nutzern des Angebots auf Verständlichkeit überprüfen zu lassen.

Tipp: Sie können schon an Hand einiger simpler Tests wie Melden Sie Ihren Hund um, Bestellen Sie ein Produkt X oder Versuchen Sie Informationen über XYZ zu finden mit wenigen Testpersonen die Gebrauchstauglichkeit Ihres Angebots überprüfen lassen.

Diese Tests sollten jedoch *nie* von Personen durchgeführt werden, die in irgendeiner Weise mit der Entwicklung des Angebots oder der anbietenden Institution in Verbindung stehen. Nur so können Formulierungen gefunden werden, die dem "Insider" vielleicht geläufig sind, für alle anderen aber ein Buch mit sieben Siegeln darstellen. Je früher diese Tests in einem Projekt stattfinden, desto größer ist die Wahrscheinlichkeit, die gewonnenen Erkenntnisse noch kostenneutral umsetzen zu können.

Ein Beispiel:

Wie viele Kommunen bietet auch die [Stadt Bonn](#) die Möglichkeit, Behördengänge auf elektronischem Wege zu erledigen. Die ersten Hürden werden hier aber bereits in der Navigation aufgebaut, da die einzelnen Navigationspunkte im besten Amtsdeutsch getextet sind.

Ob nun das vom Bürger gewünschte Antragsformular eine Lebenslage oder ein Anliegen darstellt, wird weder durch die Navigation, noch durch die Folgeseiten erklärt. Auf den Folgeseiten wird lediglich eine aus den Buchstaben des Alphabets gebildete Liste gezeigt. Über diese muß der Bürger dann erraten, mit welchem Anfangsbuchstaben das für sein Anliegen oder seine Lebenslage zuständige Amt anfängt (vorausgesetzt, er oder sie hat JavaScript aktiviert – ohne JavaScript enden die Folgeseiten in einer Sackgasse).



Für den Benutzer, der sich tatsächlich mit der Struktur der Bonner Ämter beschäftigen will, und vielen wird angesichts der Struktur dieser

Website nichts anderes übrigbleiben, gibt es ein [Organigramm des Rathauses im PDF-Format](#).

Bedingung 14.2	Prio. 2	Text ist mit graphischen oder Audio-Präsentationen zu ergänzen, sofern dies das Verständnis der angebotenen Information fördert.
Bedingung 14.3	Prio. 2	Der gewählte Präsentationsstil ist durchgängig beizubehalten.

Was heißt das?

Die grafische Gestaltung ist ein integraler Bestandteil des gesamten Web-Angebotes und dient nicht nur der Selbstdarstellung des Anbieters. Sowohl der Besucher als auch der

Anbieter haben ein Recht auf Gestaltung, aber Anbieter haben sogar die besondere Verantwortung und Pflicht, nur gut gestaltete und benutzbare Seiten zu veröffentlichen. Die BITV schließt hierbei Sonderwege wie reine Textversionen ausdrücklich aus, da auch diese nicht Barrierefrei im Sinne der Verordnung für die Gesamtheit der Besucher sein können.



Besonders für Menschen mit einer unterdurchschnittlichen Sprachkompetenz sind grafische Hilfen oftmals ein entscheidender Faktor für die erfolgreiche Bewältigung der selbstgestellten Aufgabe, für deren Erledigung diese Menschen auf Ihre Seiten gekommen sind. Hierunter fallen nicht nur Menschen mit Lernbehinderungen oder Leseschwäche, auch Migranten mit geringen Deutschkenntnissen, Besucher aus dem Ausland oder Menschen aus dem von Soziologen so genannten "bildungsfernen Milieu" fallen hierunter.



Ein Beispiel:

Tests haben gezeigt, dass viele Icons und Symbole von Besuchern nicht verstanden werden, wenn die verwendete Symbolik nicht aus ihrem Kulturraum stammt oder nicht ihren Sehgewohnheiten entspricht. Der vielfach für einen E-Mail-Link verwendete amerikanische Blech-Briefkasten ist hier ein klassisches Beispiel: Sie können nicht davon ausgehen, dass alle Besucher überhaupt verstehen, dass es sich hierbei um einen Briefkasten handelt, geschweige denn dass die dahinter liegende Funktion erkannt wird.

Wie geht es weiter?

Wie der Prozeß der Barrierefreiheit bis zu der von der [BITV](#) gesetzten Ziellinie am 31. Dezember 2005 weiter verlaufen wird, kann natürlich niemand vorhersehen. Einige Punkte, die von der Verordnung verlangt werden, aber problematisch in der Umsetzung sind, haben wir in den vergangenen drei Wochen ja bereits angesprochen. Andere Punkte, die eigentlich zu jedem professionellen Webangebot gehören, sind mittlerweile zum Standardmerkmal neuer Websites geworden.

Sie können sich jedoch sicher sein, daß das Team von 'Einfach für Alle' diesen Prozeß kritisch begleiten und an geeigneter Stelle Denkanstöße geben wird. Der seit dem 5. Mai ausgeschriebene [BIENE-Award](#) soll dazu beitragen, die Forderung nach dem Abbau von Barrieren an die Entscheider und Entwickler heranzutragen.

Wenn Sie Anregungen, Kommentare oder Verbesserungsvorschläge zu den einzelnen Artikeln haben, oder als Benutzer assistiver Werkzeuge ihre persönlichen Erfahrungen mit Barrieren im Internet mitteilen möchten, benutzen Sie bitte die [Kommentarfunktionen](#) der einzelnen Artikel.