

Accessibility Design

Um ein Höchstmaß an Web-Accessibility zu erreichen ist interdisziplinäre Zusammenarbeit gefragt – vom ersten Wireframe, über das Design und die Frontend-Webentwicklung, bis hin zu einer nachhaltigen Barrierefreiheit durch das geschulte Redaktionsteam. Dieser Artikel soll Ihnen dabei helfen, zentrale Fehler schon in der Designphase zu vermeiden.

Wer ist eigentlich für Barrierefreiheit im Internet verantwortlich? Sind es die Konzeptioner und Texter oder die User Experience und Interface-Designer? Sind es die Webdeveloper und Programmierer? Oder am Ende doch die Kunden und das spätere Redaktionsteam? Die Antwort ist: Alle genannten müssen von Anfang an Barrierefreiheit im Blick haben und Hand in Hand arbeiten. Klingt banal, ist aber in der Realität häufig nicht der Fall. Im Gegenteil, die einzelnen Abteilungen und Disziplinen scheinen sich eher stets auf den anderen zu verlassen. In Ausschreibungen der Öffentlichen Hand ist die Anforderung Barrierefreiheit entsprechend der BITV mittlerweile fester Bestandteil. Selten finden Sie in Ausschreibungen jedoch eine Aussage darüber, ob und wie das Ergebnis später daraufhin überprüft wird.

Barrierefreiheit im Internet ist weitestgehend ein Technik-getriebenes Thema. Webstandards, HTML5-Semantik, Progressive Enhancement oder Graceful Degradation, versteckte Screenreader-Hilfen, ARIA und dergleichen mehr, stehen stets im Fokus. Accessibility Design (so nennen wir es jetzt mal) hingegen nicht. Dabei können falsche Design-Entscheidungen im späteren Projektverlauf häufig nur schwer wieder rückgängig gemacht werden – zumindest nicht unauffällig. Denn wenn ein Design erstmal freigegeben ist, möchten in der Regel weder Auftraggeber, noch Auftragnehmer beispielsweise Farbkontraste oder Schriftarten und Schriftgrößen neu abstimmen.

Barrierefreiheit mit Auge

Die BITV (Barrierefreie-Informationstechnik-Verordnung) ist in Deutschland DAS Regelwerk, wenn es um die Überprüfbarkeit von Barrierefreiheit im Internet geht. Auf internationaler Ebene sind es die WCAG (Web Content Accessibility Guidelines). Dieser Artikel klopft aber nicht einfach nur alle Prüfschritte der BITV nacheinander ab und wendet sie auf den Designprozess an. Hier geht es um Barrierefreiheit mit Auge: BITV-Experten werden beispielsweise einwenden, dass das Thema Ladezeiten (Performance) kein expliziter Prüfschritt der BITV ist. Trotzdem ist das Thema Ladezeiten vor dem Hintergrund der stetig wachsenden mobilen Nutzung unbestreitbar ein hochaktuelles Thema. Und außerhalb performanter WLAN-Netzwerke sind langsame Seiten für alle Menschen eine schwere Barriere. Oder um es anders auszudrücken: Wenn Sie alle Regeln der BITV berücksichtigen, aber trotzdem eine Seite mit 10 MB ausliefern, ist ihre Seite außerhalb von DSL de facto unzugänglich.

#a11y Design im Überblick

Aspekte der Barrierefreiheit, die bereits in der Designphase relevant sind, beinhalten:

- Struktur und Navigation
- Farben und Kontraste (auch für funktionale Grafiken)
- Alpha-Transparenz (in Verbindung mit Text und wechselnden Hintergründen)
- Schriftgrößen (Schriftvergrößerung)
- Schriftarten und Schriftauszeichnung (Lesbarkeit)
- Pseudoklassen :hover, :focus, :active (Tastatursteuerung)
- Reihenfolge der Inhalte (auf verschiedenen Endgeräten)
- (Übersprung-)Navigations-Mechanismen & Brotkrumenpfad
- Slider (Besonderheit zeitgesteuerter Inhalte)
- Bilder mit Bildunterschriften, Überschriften und umgebenden Text (und dessen Bedeutung für Alternativtexte)

- Formulare und Fehlermeldungen
- HTML5 Semantik (vorausgedacht)

Struktur und Navigation

Ohne verbindliche Struktur und ohne benutzerorientiertes Navigationskonzept laufen viele Aspekte der Barrierefreiheit erfahrungsgemäß ins Leere. Zu viele Hauptnavigationen erschweren beispielsweise später eine Skalierbarkeit der Textinhalte bei einer Monitorauflösung von 1024 Pixeln Breite. Zu lange Begriffe innerhalb der Navigation insgesamt führen später fast unweigerlich zu kaum lesbaren Brotkrumenpfaden. (Ariadne-Pfad). Versuchen Sie bereits in der Designphase drei zentrale Fragen des Benutzers zu beantworten: „wo bin ich, wo war ich und wo kann ich hingehen?“. Bedenken Sie diese Frage nicht nur für die Startseite, sondern auch für Unterseiten, auf die Besucher in der Regel viel häufiger über Suchmaschinen stoßen. Achten Sie dabei nicht nur auf Benutzer, die Ihrem visuellen Designkonzept folgen können, sondern erleichtern Sie auch blinden Nutzern die Navigation und Orientierung durch alternative Zugangsmechanismen, wie Suchfunktion, Sitemap oder Themen-Cluster.

Farben und Kontraste

Wichtig: Verwenden Sie niemals Farbe als einziges Mittel, um wichtige Informationen (zum Beispiel eine Warnung) darzustellen. Das gilt beispielsweise auch für Fehlermeldungen innerhalb von Formularen. Neben der farblichen Differenzierung spielt das Kontrastverhältnis von Vorder- zu Hintergrundfarbe eine entscheidende Rolle. Die Vorgaben für das Kontrastverhältnis sind 4,5:1 sowie mindestens 3:1 für große Schrift (ab 18,7 Pixel fett bzw. 24 Pixel normal gilt der Wert für „Großen Text“). Das Kontrastverhältnis müssen Sie auch für funktionale Grafiken beachten (beispielsweise für das so genannte Burger-Menü).

Alpha-Transparenz

Alpha-Transparenz vor wechselnden Hintergründen spielt in Verbindung mit der Einhaltung von Kontrastvorgaben eine Rolle. Wenn Sie beispielsweise Bilder mit einer halbtransparenten Fläche überlagern, auf der Sie Text positionieren, sollten Sie den Transparenzwert der Farbfläche sowohl mit dunklen Bildern, als auch mit hellen Bildern testen. Dies ist insbesondere dann wichtig, wenn die Hoheit über die Bildauswahl später beim Redaktionsteam liegt. Selbst wenn Sie dunkle Bilder vorgesehen haben, können Sie mit Sicherheit davon ausgehen, dass später auch helle Bilder als Hintergrund ausgewählt werden.

Tipp: Legen Sie einfach zu Testzwecken eine schwarze und eine weiße Fläche hinter den halbtransparenten Farbbalken. Wenn in beiden Fällen die Farbwerte für den Text auf der halbtransparenten Farbfläche den Kontrastvorgaben entsprechen, sind Sie auf der sicheren Seite.

Schriftgröße (Schriftvergrößerung)

Die Schriftgröße ist vor allem in Bezug auf die bereits erwähnte Forderung nach ausreichend Kontrast zwischen Vorder- und Hintergrundfarbe von Bedeutung. Wichtig ist aber auch, dass Sie für das gesamte Design von einer Minimal-Textgröße (Basis-Schriftgröße) von 14 Punkt ausgehen. Große Schriftgrößen (zum Beispiel Überschriften) bauen darauf auf und sind entsprechend proportional größer. Mehr Informationen finden Sie dazu auch beim W3C.

Ein Aspekt, der in der Designphase häufig vergessen wird – später aber kaum mehr abzufedern ist – ist die Forderung nach zweihundertprozentiger Schriftvergrößerung. Hierbei geht es nicht um den (in allen Browsern mittlerweile gängigen) Seiten-Zoom, sondern explizit um reine Textvergrößerung. Im Contentbereich ist das in der Regel kein größeres Problem. Komplexe Navigationen und interaktive Elemente, wie beispielsweise Text-Bild-Slider, können hingegen schnell unbenutzbar werden, wenn sich Elemente durch den Text-Zoom überlagern. Hier hilft nur eins: Nutzen

Sie den zur Verfügung stehenden Raum im Design nicht vollständig aus, sondern geben Sie Elementen Platz zum Wachsen.

Schriftarten und Schriftauszeichnung

Wenn es um Barrierefreiheit geht, müssen immer verschiedene Nutzergruppen berücksichtigt werden. Die technische Frontend-Entwicklung agiert de facto vollständig „unter der Haube“ und adressiert zu einem sehr großen Prozentsatz Menschen, die auf einen Screenreader angewiesen sind. Egal, ob es sich um HTML5-Semantik, Überschriften-Outline, Tastatursteuerung, Übersprung-Mechanismen, versteckte Hilfetexte oder ARIA-Anreicherung handelt. Das Accessibility Design hingegen zielt schwerpunktmäßig auf den visuellen Output. Aber Sie sollten nicht nur „Otto Normal-Nutzer“ im Hinterkopf haben, sondern auch Menschen mit Sehbehinderung, kognitiven Einschränkungen oder Leseschwierigkeiten. Letzteres betrifft ja nicht nur Menschen mit Dyslexie, sondern auch Menschen, deren Muttersprache nicht Deutsch ist.

Neben der Auswahl von Schriftart und Schriftschnitt (Condensed, Thin, Light, Regular, Medium, Bold, etc.) unterstützt auch der umsichtige Umgang mit Schriftauszeichnung (Caps, Smallcaps, Italic, etc.) die Lesbarkeit von Texten. Wenn es keine Corporate-Font-Vorgaben gibt, gehen Sie der Einfachheit halber erstmal von Schriftarten aus, die nachgewiesenermaßen auch für Menschen mit Dyslexie gut lesbar sind (Arial, Comic Sans, Century Gothic, Verdana, Trebuchet, Calibri). Falls Ihnen jetzt das Designerherz in die Hose gerutscht ist, es geht nicht darum, genau diese Schriften zu verwenden. Es geht darum, andere Schriftarten mit diesen „Referenz-Schriftarten“ zu vergleichen.

Tipps für Versalschreibweise (Caps):

Es ist festzustellen, dass Designer gerne Versalschreibweise einsetzen – häufig für Überschriften, Navigationen, Fehlermeldungen oder auch Blockquotes. Wenn Sie nicht darauf verzichten möchten, setzen Sie diese Schriftauszeichnung mit Augenmaß ein. Machen Sie den Einsatz von Versalschreibweise vom Inhalt abhängig. Besteht beispielsweise die Hauptnavigation aus den Begriffen „AKTUELLES, UNTERNEHMEN, TEAM, PARTNER“, dann ist Versalschreibweise (aus meiner Sicht) noch vertretbar, in Anbetracht der kurzen Wörter.

Verzichten Sie aber auf Versalschreibweise bei langen Begriffen, oder wenn Sie jeweils mehr als ein Wort auszeichnen wollen. Zeichnen Sie nur sehr einfache und bekannte Worte in dieser Art und Weise aus. Setzen Sie Versalschrift niemals dort ein, wo später Redakteure Inhalte mit dieser Formatierung pflegen müssen.

Kursivschrift (italic):

Auf Kursivschrift sollten Sie aufgrund der vergleichsweise schlechten Lesbarkeit am Monitor verzichten. Früher wurde Kursivschrift häufig für die Pseudoklasse `:visited` eingesetzt. Aus Datenschutzgründen ist der Einsatzbereich dieser Pseudoklasse allerdings mittlerweile ohnehin begrenzt.

Unterstreichung:

Auch auf die Unterstreichung von Texten zur visuellen Hervorhebung sollten Sie verzichten. Diese Auszeichnung ist Links vorbehalten.

Pseudoklassen `:hover`, `:focus`, `:active`

Navigation und Orientierung im Internetauftritt mithilfe der sogenannten Pseudoklassen `:hover`, `:focus` und `:active`. Oft werden „Zustände“ für interaktive Elemente im Nachhinein nicht mehr definiert, wenn dies nicht durch das Design vorgegeben wurde. Die Pseudoklassen `:hover` (mit dem Mauszeiger berührt Links/Buttons) und `:active` (mit dem Mauszeiger aktuell angeklickte Links/Buttons) dienen zur Positionsbestimmung für Anwender mit einer Computermaus. Beachten Sie auch hier die Kontrastvorgaben für Vorder- und Hintergrundfarben. Tipp: beachten Sie,

dass :hover auf Touch-Displays (Smartphones, Tablets, etc.) nicht funktioniert. Die Pseudoklasse :focus dient zur Positionsbestimmung des Tastaturfokus für Menschen, die nur mit der Tastatur als Eingabe- und Steuergerät arbeiten können. Dies sind nicht nur blinde Menschen (die mit der Computermaus als grafisches Eingabegerät nichts anfangen können), sondern auch Menschen mit motorischen Einschränkungen. Wenn die Pseudoklasse :focus nicht explizit vergeben wurde, greift immer der Default-Wert des jeweiligen Browsers. Leider reichen die Kontrastwerte dieser Default-Werte traditionell browserübergreifend nicht aus. Sie können sich also nicht darauf verlassen.

Tipp: Kneifen Sie doch mal die Augen zusammen (um starke Sehbeeinträchtigung zu simulieren) und versuchen Sie dem Default-Fokus zu folgen. In den meisten Fällen ist das unmöglich. Da der Tastatur-Fokus wirklich nur bei der Tastatursteuerung sichtbar wird (also bei Menschen, die wirklich drauf angewiesen sind), sollten Sie in Bezug auf Sichtbarkeit auch nicht zimperlich sein. Eine Border von mindestens 2 Pixeln in einer kräftigen Farbe auf allen interaktiven Elementen (Formulare nicht vergessen) hat sich bewährt.

Reihenfolge der Inhalte

Hand aufs Herz, wie oft „denken“ Sie die Reihenfolge Ihrer Website-Inhalte linear – also so, wie die Inhalte (ohne CSS und JavaScript) im HTML stehen? Zugegeben, vermutlich werden Sie diese Frage seit ein paar Jahren (Stichwort: Responsive Webdesign auf Smartphones) unbewusst wieder zu Ihren Gunsten beantworten können. Allerdings in der Regel aus anderen Beweggründen: wenn zum Beispiel Kontakt-Informationen auf ein Smartphone nicht mehr „rechts oben“ stehen, wie in der Desktopansicht, sondern plötzlich am Seitenende.

Screenreader:

Die lineare „Smartphone-Ansicht“ (lassen wir JavaScript und Media Queries mal außen vor) kommt der Ansicht für Screenreader-Nutzer schon recht nahe – sehr vereinfacht (aber plakativ) ausgedrückt. Allerdings ohne Kontext und visuelle Orientierung. Wenn Sie eine grobe Vorstellung davon haben wollen, wie Screenreader-Nutzer Ihre Seite „sehen“, dann schalten Sie doch einfach mal CSS ab. Natürlich können Sie so nur eine lineare Ansicht simulieren. HTML-Semantik, versteckte Hilfetexte sowie gängige und auch spezielle HTML-Attribute, die der Screenreader-Orientierung dienen, werden dabei nicht berücksichtigt. Wenn Sie ein realistisches Bild aus der Perspektive eines Screenreader-Nutzers bekommen wollen, dann installieren Sie sich doch mal den kostenlosen NVDA Screenreader oder greifen auf die integrierten Screenreader auf Ihrem mobilen Endgerät zurück. Android bietet dafür Talkback. Apples iOS hält Voiceover bereit.

Skip-Links:

Die Reihenfolge von Inhalten, in Bezug auf ihre Bedeutung, sollten Sie von vornherein konzeptionell berücksichtigen. UX-Design ist in der Regel Kontext-basiert. Inhaltliche Gewichtung findet mithilfe von Aussehen (zum Beispiel Piktogrammen), Positionierung, Größe und farblicher Differenzierung statt. Wichtige Elemente sind zumeist größer, zusätzlich hervorgehoben und befinden sich oft „above the fold“. Unwichtigere Elemente treten zurück. Bedenken Sie, dass dies ein rein visuelles Konzept darstellt. Blinde Menschen müssen sich Internetseiten auf andere Weise erschließen. Dazu zählen neben sprechenden URLs, aussagekräftigen Seitentiteln, sinnvollen Überschriften-Outlines (H1 bis H6), HTML5-Semantik (respektive ARIA Landmark-Roles) und Listen-Navigtionen auch alternative Navigations-Mechanismen wie eine Seitensuche, Sitemap und Brotkrumenpfad. Dazu zählen aber auch Skip-Links (versteckte Navigations-Elemente, die nur sichtbar werden, wenn man die Tastatursteuerung verwendet), um große Contentblöcke zu überspringen und direkt zur Hauptnavigation oder zum Content zu gelangen. Aber auch hier gilt wieder: lassen Sie Augenmaß walten. Mehr als drei Skip-Links schießen über das Ziel hinaus. Auch diese müssen Sie gestalten. Ansonsten wissen Ihre Frontend-Developer nicht, wie sie aussehen und wo sie positioniert sein sollen.

Slider (Zeitgesteuerte Inhalte)

Zeitgesteuerte Inhalte, wie beispielweise Slider müssen durch den Nutzer angehalten und abgespielt werden können. Sehen Sie in Ihrem Design für derartige dynamische Elemente einen Start-Stopp-Button vor. Beachten Sie auch hier die Kontraste für die Funktionselemente.

Ladezeiten

Schon in der Designphase können Sie die Grundlage für eine verbesserte Performance legen.

Bildauswahl:

Vor allem große Bilder verzögern das Laden einer Webseite. Insbesondere dann, wenn die redaktionelle Hoheit über die Bildkompression später bei Redakteuren liegt, die in Fragen der optimalen Bildkompression unerfahren sind. Das ist in der Regel bei Bild-Slidern der Fall. Versuchen Sie bei der Bildkonzeption bereits so viele Kilobyte wie möglich zu sparen. Makrobilder, Bildausschnitt und Close-Ups mit weniger Bilddetails und viel Hintergrundunschärfe bewirken Wunder. Versuchen Sie in der Designphase von vornherein auf die Anzahl der Bilder einzuwirken. Ein normaler Text-Bild-Slider mit nur fünf Motiven von je 50 Kilobyte wiegt inklusive HTML, CSS und JavaScript erfahrungsgemäß schon um die 275 Kilobyte. Behalten Sie von Anfang an für zentrale Einstiegsseiten das Gesamtvolumen von Code und Assets im Blick.

@Fontface:

Auch die Wahl der richtigen Schriftart kann sich positiv auf die Ladezeiten auswirken. Wenn Sie nicht auf Systemschriftarten zurückgreifen wollen, sollten Sie auf eine schlanke Auswahl der Google Font API setzen. Nutzen Sie das Google Fonts Page Load Tool, um das „Download-Gewicht“ Ihrer Schriftauswahl zu testen. Versuchen Sie im wahrsten Sinne des Wortes im „grünen Bereich“ zu bleiben. Linken Sie am besten direkt auf die Google Font API ([href='https://fonts.googleapis.com/css?family...'](https://fonts.googleapis.com/css?family...)). Aber das ist etwas für die Webentwickler, nicht für den Designer. Wenn Sie als Designer nicht auf Systemschriftarten setzen, oder sich bei der Google Font API bedienen, bleibt Ihnen nur die Möglichkeit Schrift über @fontface einzubinden. Je nachdem für welche Schriftart Sie sich im Design entscheiden (und für wie viele Schriftschnitte), kann das @fontface Paket mit mehreren 100 Kilobyte aber eine beachtliche Größe erlangen.

SVG:

Wenn Sie grafische Elemente in Ihrer Seite einbauen, dann versuchen Sie von vornherein Vektorgrafiken zu nutzen. Stellen Sie Ihren Workflow daraufhin ab, egal welches Programm Sie für die Designentwicklung verwenden.

Icon-Fonts:

Auch wenn Icon-Fonts im gesamten Workflow sicherlich Vorteile mit sich bringen und Icon-Fonts sich auch auf Ladezeiten positiv auswirken können, in Bezug auf Barrierefreiheit sind sie mit großer Vorsicht zu genießen. Wenn Sie (und Ihr Entwicklerteam) nicht genauestens mit den Problemen vertraut sind, die mit Icon-Fonts einhergehen können, nehmen Sie lieber Abstand davon.

Bilder mit Bildunterschriften

Alternative Texte für Nicht-Textelemente stellen Redakteure immer wieder vor große Herausforderungen. Gleiches gilt übrigens für sinnvolle title-Attribute. Im Gegensatz beispielsweise zu Farbkontrasten, die sich mit Prüftools leicht testen lassen, gibt es für Alternativtexte kein „100 Prozent richtig“, wohl aber ein „100 Prozent falsch“.

Alt="6846asdf65749.jpg" oder Alt="Logo" sind beispielsweise keine guten Alternativ-Texte. Eine gute Beispiel-Übersicht haben die Pennsylvania State University und WebAIM zusammengestellt. Das Fazit von WebAIM sollten Sie sich merken: „Adding alternative text to images is one of the easiest accessibility principles to learn and one of the hardest to master“. Unterstützen Sie Redakteure bereits in der Designphase mit gut durchdachten Bild-Text-Konzepten. Insbesondere, wenn Sie mit dem Gedanken spielen, Bildunterschriften als visuellen Akzent einzusetzen.

Bilder mit Caption:

Bildunterschriften sind keine Neuheit, haben aber mit den semantischen HTML5 Elementen <figure> und <figcaption> nochmal mehr Bedeutung erlangt. Aus typografischer Sicht gestalterisch reizvoll, sollten Sie sich aber über Implikationen hinsichtlich der Barrierefreiheit bewusst sein. Wir bewegen uns hier im Spannungsfeld von „too much accessibility“. Die Frage lautet, was im Alt-Attribut stehen muss, wenn ein Bild eine Caption aufweist? Bruce Lawson hat dazu mal ein interessantes Quiz auf html5doctor.com veröffentlicht, das auch reichlich kommentiert wurde. Richtig kompliziert wird es, wenn derartige Bild-Bildunterschrift-Kombination noch von einer Überschrift umrahmt wird. Dann sollten Sie Ihr Konzept nicht nur mit Blindtext testen, sondern vollständig durchspielen – inklusive Alt-Attribut (auch wenn dieses in Ihrem Layout nicht sichtbar ist).

Formulare und Fehlermeldungen

Die korrekte „Gestaltung“ von Formularen und Fehlermeldungen klingt vielleicht mehr nach einer Aufgabe für Frontentwickler, die Weichen werden aber im Design gestellt. Wenn Sie beispielsweise aus gestalterischen Erwägungen das Legend-Element weglassen, oder ein Suchfeld aus Platzgründen kein Label bekommt. Oder wenn Sie wichtige Informationen, wie „Pflichtfelder sind mit einem Sternchen gekennzeichnet“ aufgrund visueller Überlegungen am Ende eines Formulars platzieren. Oder wenn Sie Formularfelder vorbelegen, aber für ein harmonisches Gesamtbild einen zu geringen Farbkontrast wählen. Oder wenn Hilfetexte aus Platzmangel nur über einen Button zugänglich sind und über einen Layer eingeblendet werden. Bedenken Sie dann auch ein entsprechendes JavaScript-Fallback. Verlassen Sie sich nicht darauf, dass Ihr Frontentwickler das schon auf dem Schirm haben wird.

Fehlermeldungen:

Wie sehen eigentlich Fehlermeldungen aus und an welcher Stelle im Formular werden sie genau ausgegeben? Berücksichtigen Sie hier nicht nur ausreichend Farbkontraste. Beachten Sie auch, dass derart wichtige Informationen nicht nur über Farbe alleine kenntlich gemacht werden dürfen (Klassiker: alle rot umrandeten Felder sind Pflichtfelder). Konvertieren Sie Ihr Formular-Design inklusive der Fehlermeldungen gelegentlich mal in ein Graustufenbild. Wenn Fehlermeldungen dann immer noch eindeutig erkennbar und gut zuzuordnen sind, haben Sie im Design Ihre Hausaufgaben gemacht.

HTML5-Semantik

Wie eingangs bereits festgestellt, Barrierefreiheit im Internet ist weitestgehend ein Technik-getriebenes Thema. Deshalb richtet sich dieser Artikel bewusst an Designer mit begrenztem HTML-, CSS- und JavaScript-Knowhow. Auch wenn die Vorstellung, dass Designer das Web gestalten, manchen Webdeveloper nicht gefällt: Es ist in weiten Teilen

die Realität. Und dieser Artikel soll dabei helfen das Web ein bisschen zugänglicher zu gestalten.

Semantische HTML5-Elemente:

Auch wenn Sie als Screendesigner möglicherweise nicht viele HTML-Spezifikationen kennen, sollten Sie sich mit ein paar semantischen HTML5-Elementen vertraut machen. Zum Glück sind semantische HTML5-Elemente teilweise selbsterklärend – zumindest hinsichtlich Ihres grundsätzlichen Einsatzzweckes. Beispielsweise: section, article, aside, header, footer, nav, figure, figcaption, address, oder time. Für den Designer (und auch später für den Webentwickler) ist es von Vorteil, wenn das Layout bzw. die Seitenstruktur von vorneherein gedanklich in die genannten HTML5-Bereiche unterteilt wird. Das setzt aber beispielsweise voraus, dass Sie wissen, wie die Elemente auch in komplexeren Layouts ineinandergreifen. Beispielsweise muss das section-Element immer mit einem Überschriften-Element eingeleitet werden. Wenn Sie dies im Design nicht berücksichtigen, können Sie entweder einen Bereich nicht mit diesem Element auszeichnen, oder der W3C-Validator wird das HTML nicht durchwinken.

Fazit

Für ein Höchstmaß an Barrierefreiheit müssen viele Disziplinen zusammenarbeiten. Nicht alle Projektbeteiligten sind Accessibility-Experten oder kennen die Forderungen der BITV. Die Technikfraktion kann Konzeptions- und Designfehler nicht persé ausbügeln. Und Online-Redakteure brauchen in allen Belangen Unterstützung. Sei es durch Schulung, oder durch Konzepte, die Redakteure dort abholen, wo sie stehen. Designer und Konzeptioner können bereits in der Entwicklungsphase von Webprojekten positiv auf Ergebnis der Barrierefreiheit einwirken.

Text: Jörg Morsbach, Geschäftsführer und kreativer Ideengeber der auf Barrierefreiheit spezialisierten Düsseldorfer Agentur anatom5. (Erschienen im Screenguide Magazin Nr. 29)