Barrierefreies Internet

Gesetze und Richtlinien

ICIHD/ICF

The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect. **Accessibility awareness**: Everyone should be accommodated, especially when around 20 percent of the population have special requirements. Nearly 50 percent of people need to make some sort of adjustment to their system to interact with it.

Impairment: any loss or abnormality of psychological, physiological or anatomical structure or function.

Disability: any restriction or lack (resulting from an impairment) of ability to perform an activity in the manner or within the range considered normal for a human being.

Handicap: a disadvantage for a given individual that limits or prevents the fulfillment of a role that is normal.

Behinderung ist die Differenz (die **„Lücke**“) zwischen **Fähigkeit** einer Person und **Anforderung** der Umwelt

Political correctness
– Vermeide Substantivierung -> Menschen mit Behinderung, behinderte Menschen
– Vermeide „behindertengerecht“ -> Barrierefrei (zugänglich/accessible)

diskriminierende Sprache
Behinderte, Menschen mit besonderen Bedürfnissen, Taubstumm, Zwerg, Liliputaner, Idiot, Depp, geistig Behindert, Mongoloid

Behinderungsarten

- Einschränkungen in der Sehleistung – Fehlsichtigkeit, Blindheit, Farbfehlsichtigkeit
- Einschränkungen in der Hörleistung – Gehörlos, schwerhörig
- Motorische Einschränkungen – Obere Extremitäten, untere Extremitäten
- Intellektuelle und psychische Beeinträchtigungen
- Technikfern
- Altersbedingte Einschränkungen – Hören, Sehen, Motorik,Reaktionsgeschwindigkeit

Die Menschenrechte

Universal Deklaration – UNO Resolution 217A (III) vom 10.12.1948

Artikel 1 - Alle Menschen sind frei und gleich an Würde und Rechten geboren.

Artikel 19 - Jeder hat das Recht auf Meinungsfreiheit und freie Meinungsäußerung.

Artikel 22 - Jeder hat als Mitglied der Gesellschaft das Recht auf soziale Sicherheit und Anspruch darauf, **wirtschaftlichen, sozialen und kulturellen Rechte zu gelangen**.

Artikel 26
1. Jeder hat das Recht auf Bildung.
2. Die Bildung muss auf die volle Entfaltung der menschlichen Persönlichkeit und auf die Stärkung der Achtung vor den Menschenrechten und Grundfreiheiten gerichtet sein.

UN Standardregeln

UN Standardregeln (1993) - Standardregeln zur Chancengleichheit (22 Regeln)

Although not a legally binding instrument, the Standard Rules represent a strong moral and political commitment of Governments to take action to attain equalization of opportunities for persons with disabilities. The rules serve as an instrument for policy-making and as a basis for technical and economic cooperation The Rules incorporate the human rights perspective which had developed during the Decade. The 22 rules concerning disabled persons consist of four chapters - preconditions for equal participation, target areas for equal participation, implementation measures, and the monitoring mechanism - and cover all aspects of life of disabled persons.

Rule 1) Sensibilisierung der Allgemeinheit (Awareness-raising) – Die Staaten sollen Maßnahmen ergreifen, um der Gesellschaft die Lage der Menschen mit Behinderungen, ihre Rechte, ihre Bedürfnisse und ihren Beitrag stärker bewusst zu machen.

Rule 5) Barrierefreie Umwelt (Accessibility) – Die Staaten sollen bei der Herstellung der Chancengleichheit in allen Gesellschaftsbereichen die allgemeine Bedeutung einer barrierefreien Umwelt erkennen. (a) Aktionsprogramme (b) Maßnahmen ergreifen

Rule 6) Bildung (Education) – Die Staaten sollen das Prinzip der Chancengleichheit für behinderte Kinder, Jugendliche und Erwachsene in den Grundschulen, weiterführenden Schulen und im Hochschulbereich in einem integrativen Umfeld anerkennen.

UN Standardregeln

Regelung in den USA - Section 508 (mit WCAG 2 AA heutzutage)

Rehabilitation Act of 1973 – verbietet Diskriminierung aufgrund von Behinderung

Anforderung an Informationstechnik - Section 508 wurde 1998 „neu“ in den US Rehabilitation Act aufgenommen: **Angestellte der Bundeseinrichtung und Öffentlichkeit**

Section 508 muss eingehalten werden von **Bundesbehörden UND Zulieferer von IT** (HW und SW) -> großer Einfluss auf Wirtschaft weil ein sehr großer Auftraggeber die Bedingungen festlegt.

Hauptbereiche von Section 508
– Anwendungssoftware und Betriebssysteme (benutzerfreundliche Anwendungen, alternative Bedienmöglichkeiten...)
– Webanwendungen (barrierefreier Zugang zu Webseiten, Screenreader...)
– Telekommunikationsprodukte (Hörbehinderung...)
– Videos und Multimedia (Untertitelung...)
– Eigenständige Produkte z.B. Drucker, Kopierer (Bedienbarkeit mit Hilfsmitteln)
– Desktop und Portable Computer

1194.21 Software *Anwendungen und Betriebssysteme*
– Funktionen mit Tastatur ausführen
– Accessibility Funktionalität anderer Produkte darf nicht unterbrochen oder ausgeschaltet werden
– Klarer Input-focus
– Interface Elemente klar identifizierbar, Statusanzeigen
– Bedeutung von Bildern konsistent und eindeutig
– Textuelle Information über OS-Funktionen
– Anwendungen dürfen benutzerdefinierte Anzeigeeinstellungen nicht ändern
– Animationen müssen auch in einer anderer (nicht animierten) Modalität präsentiert werden
– Farbe darf nicht der einzige Informationsträger sein
– Wenn ein Produkt Farbeinstellung zulässt, dann soll eine Vielzahl von Wahlmöglichkeiten geboten werden
– Keine Blinkfrequenzen zwischen 2 und 55Hz
– Elektronische Formulare müssen auch mit Hilfsmitteln bedient werden können

UN Behindertenrechtskonvention

Übereinkommen über die Rechte von Menschen mit Behinderungen

Behindertenrechtskonvention, BRK) Austria in 2008

– Völkerrechtlicher Vertrag (weltweit bindendes Rechtsinstrument)

— eine Vielzahl spezieller, auf die Lebenssituation behinderter Menschen abgestimmte Regelungen.

– Gleichberechtigte Teilhabe und Teilnahme von ‚Menschen mit Behinderung' am gesellschaftlichen Leben.
– Keine genaue und abschließende Definition des Begriffs Behinderung, sondern ein Verständnis von "Behinderung"
– langfristige körperliche, seelische, geistige oder Sinnesbeeinträchtigungen .... welche am vollen und gleichberechtigten Gebrauch der fundamentalen Rechte hindern.
– Die BRK orientiert sich am sozialen Verständnis von Behinderung.
– Umfassende Strategien der Staatlichen Behindertenpolitik
– Ziel ist eine inklusive Gesellschaft
– Integration - Inklusion; Teilhabe , Teilnahme
– Monitoringausschuss zur Überwachung der Umsetzung (Experte, Organisationen, Vertreter)

Gesetze und Richtlinien in Österreich

Österr. Bundesverfassung Art. 7, Abs. 1: („Antidiskriminierungsparagraph") – Alle Staatsbürger sind vor dem Gesetz gleich

Artikel 8 des Bundesverfassungsgesetzes wurde um folgende Sätze ergänzt: (2005) – „Die Österreichische Gebärdensprache (ÖGS) ist als eigenständige Sprache anerkannt.

Bundesgesetz über die Gleichstellung von Menschen mit Behinderungen (BundesBehindertengleichstellungsgesetz – BGStG 1.1.2006) – § 1. Ziel dieses Bundesgesetzes ist es, die Diskriminierung von Menschen mit Behinderungen zu beseitigen oder zu verhindern und damit die gleichberechtigte Teilhabe von Menschen mit Behinderungen am Leben in der Gesellschaft zu gewährleisten und ihnen eine selbstbestimmte Lebensführung zu ermöglichen.

BGstG definiert Begriffe:

– **Behinderung**: § 3. Behinderung im Sinne dieses Bundesgesetzes ist die Auswirkung einer nicht nur vorübergehenden körperlichen, geistigen oder psychischen Funktionsbeeinträchtigung oder Beeinträchtigung der Sinnesfunktionen, die geeignet ist, die Teilhabe am Leben in der Gesellschaft zu erschweren. Als nicht nur vorübergehend gilt ein Zeitraum von mehr als voraussichtlich sechs Monaten.

– **Diskriminierungsverbot**: § 4. (1) Auf Grund einer Behinderung darf niemand unmittelbar oder mittelbar diskriminiert werden. (2) Eine Diskriminierung liegt auch vor, wenn eine Person auf Grund ihres Naheverhältnisses zu einer Person wegen deren Behinderung

diskriminiert wird.
§ 5. (1) *Eine unmittelbare Diskriminierung liegt vor*, wenn eine Person auf Grund einer Behinderung in einer vergleichbaren Situation eine weniger günstige Behandlung erfährt … Anschein nach neutrale Vorschriften, Kriterien oder Verfahren sowie Merkmale gestalteter Lebensbereiche Menschen mit Behinderungen …. bei Anweisung einer Person zur Diskriminierung aus dem Grund einer Behinderung vor … Belästigung liegt vor, wenn im Zusammenhang mit einer Behinderung eine unerwünschte Verhaltensweise gesetzt wird, 1. die die Würde der betroffenen Person verletzt oder dies bezweckt, 2. die für die betroffene Person unerwünscht, unangebracht oder anstößig ist und 3. die ein einschüchterndes, feindseliges, entwürdigendes, beleidigendes oder demütigendes Umfeld für die betroffene Person schafft oder dies bezweckt … 1. bei Anweisung zur Belästigung einer Person, 2. wenn die Zurückweisung oder Duldung einer Belästigung durch die belästigte Person zur Grundlage einer diese Person berührenden Entscheidung gemacht wird, 3. wenn eine Person auf Grund ihres Naheverhältnisses zu einer Person wegen deren Behinderung belästigt wird.

– **Unverhältnismäßige Belastungen**: § 6. (5) Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstände, Systeme der Informationsverarbeitung sowie andere gestaltete Lebensbereiche, wenn sie für Menschen mit Behinderungen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind.

„Das BGStG tritt Anfang 2016 nach zehnjähriger Übergangsfrist in vollem Umfang in Kraft. “

Behinderte Menschen, die durch Barrieren diskriminiert werden, können Schadenersatz geltend machen.

Änderung im Verbandsklagsrecht seit 1.1.2018

„Wird gegen die im BGStG geregelten gesetzlichen Gebote oder Verbote verstoßen, und werden dadurch die allgemeinen Interessen von Menschen mit Behinderung wesentlich und dauerhaft beeinträchtigt, so können Klagsverband, Behindertenanwalt und Österreichischer Behindertenrat eine Klage auf Feststellung, sowie bei großen Kapitalgesellschaften im Sinne des Unternehmensgesetzbuchs (UGB) auch auf Unterlassung und Beseitigung einer Diskriminierung aus dem Grund einer Behinderung einbringen.“

Bundesgesetz über Regelungen zur Erleichterung des elektronischen Verkehrs mit öffentlichen Stellen - **E-Government-Gesetz - E-GovG** (2004, ...2018)

Gegenstand und Ziele:
– Die Wahlfreiheit zwischen Kommunikationsarten für Anbringen an die öffentliche Verwaltung.
– Die Gewährleistung von Sicherheit und Datenschutz im elektronischen Verkehr durch besondere technische Mittel.
– Behördliche Internetauftritte so gestalten, dass internationale Standards über die Webzugänglichkeit auch hinsichtlich des barrierefreien Zugangs für behinderte Menschen eingehalten werden.
– Staatliche Einrichtungen mussten bis Anfang 2008 volle Webzugänglichkeit nach internationalen Standards erreichen (war WCAG 1 A)
– WCAG 2.0 automatisch erfasst, da im Gesetz der aktuelle Stand der Technik maßgebend ist
– bundesweiter Styleguide

Bundesweiter Styleguide

Aktuelle Version 3.0 vom 31.05.2017

– E-Government Styleguide für E-Formulare
– aktueller Stand der Technik (HTML5, JavaScript, usw.)
– legt einen besonderen Fokus auf die zugängliche Gestaltung von E-Formulare der öffentlichen Verwaltung Österreichs
– Standarddaten für E-Formulare
– für die immer wiederkehrenden Grunddaten von Formularen (wie z.B. die Adressdaten) die wichtigsten Parameter (wie Bezeichnung der Felder, deren Länge, Inhalt, Typ, usw.) im Dokument. Auch eine Mustersammlung von Standard- Formularbausteinen. (z.B. Es wird beschrieben welche Formularelemente html unterstützt und dass daher NUR solche verwendet werden dürfen)
– Ausprobieren ohne Folgen für Formulare

Styleguide beinhaltet:

– die relevanten Anforderungen der Web Accessibility Guidelines 2.0 (WCAG 2.0).
– Prototypen von E-Formularen bzw. Formularelementen (Best Practice sg-visio)
– Muster basierend auf dem Bootstrap-Framework 2

– Grundsätzlicher Aufbau von E-Formularen
– Der Kopfteil beinhaltet...
– Im Teil Formularinhalte befinden sich...
– Im Fußteil sind folgende Elemente angeordnet...
Arten der Benutzermeldungen: Pflicht, Hilfe, Hinweis, Warnhinweis, Erfolgsmeldung

Normen - Standards

Eine Norm ist ein Dokument, das mit Konsens (innerhalb einer Gruppe von Experten) erstellt und von einer anerkannten Institution angenommen wurde und das für die allgemeine und wiederkehrende Anwendung Regeln, Leitlinien oder Merkmale für Tätigkeiten oder deren Ergebnisse festlegt. **Eine Norm ist kein Gesetz.** ***Ein Gesetz (oder anderes verbindliches Regelwerk) kann sich aber auf Normen beziehen.***

ÖNORM B 1600: 2013 10 01: Barrierefreies Bauen – Planungsgrundlagen
ÖNORM A 3011-3 - Graphische Symbole für die Öffentlichkeitsinformation
DIN 1450:2013-04 : Schriften - Leserlichkeit

Europäische Norm EN 301549 "*Accessibility requirements suitable for public procurement of ICT products and services in Europe*"

– Zugänglichkeitsanforderungen für öffentliche Beschaffung von ICTProdukten und Diensten Version 2.1.2 von August 2018 Ausschreibungskriterien für alle Bereiche Barrierefreier Informationstechnik. Neben Web, digitalen Dokumenten und Software auch für Hardware, wie zum Beispiel Automaten.
– Die Norm enthält alle A und AA-Kriterien der WCAG 2.1 (früher WCAG 2)

ARIA

Allgemein

HTML:

– wurde nicht entwickelt, um WebApplikationen zu erstellen.
– Ist statisch / wurde um ein sequentielles Client-Server-Modell entwickelt.
– hat nur (sehr) wenige InterfaceElemente zur Verfügung hat
– Web-Applikationen mit komplexeren Widgets mit HTML-JavaScript Mischung realisiert
-> Widget verhält sich wie eine DesktopApplikation
– Visuell sind diese komplexeren Komponenten gut zu erkennen und mit der Maus i.Allg. auch gut zu bedienen
– Aber wie geht das mit der Tastatur
– Was ist die Aufgabe des Widgets
– Was sind Status und Eigenschaft

Inhalte aktualisiert – **Assisitive Technologien 'erwarten'**, dass sich Inhalte bei einem Navigationsereignis (klick auf einen Link) ändern.

Mit Techniken wie z.B. **AJAX** (Asynchronous JavaScript and XML) können **Webinhalte 'versteckt'** (und auch nur teilweise) ausgetauscht werden – **Assistive Technologien** können das manchmal **nicht erkennen** oder wissen nicht wo es passiert ist.

WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications) ist eine Spezifikation die Hilfen anbietet diese Probleme zu lösen. (Kurzform: ARIA)

ARIA

ARIA ist eine positive Technologie die etwas ermöglicht, aber nicht verbietet!

- sehr einfach zu implementieren.
- nur im Markup aber keine ARIA zum HTML Markup - hinzufügen wenn schon die korrekte Semantik impliziert wird

Alle (großen) Browser und eine Vielzahl assistierender Technologien (z.B. Jaws, WindowEyes) unterstützen ARIA.

Wenn in einem Dokument aria role und aria-attribute verwendet wird, validiert das nicht mehr gegen XHTML (man könnte das aber selber in ein DTD schreiben)

ARIA ist in HTML5 integriert

Aria beschreibt 'Roles', 'States' und 'Properties' für Widgets und Dokumente damit sie zugänglich werden.
Erzeugt (stellt zur Verfügung) zusätzliche 'Metadaten' um 'Rich Internet Applications' zugänglich zu machen.
Erlaubt besseren Zugriff über Tastatur.
= **Eine Definition von Attributen (und dazugehörigen Werten) die man in HTML Elementen angeben kann.**

'HTML-Attribute'

ARIA-Rollen (role="value")
– Um Widgets genauer zu spezifizieren
– Um Seitenstruktur genauer zu kennzeichnen
– Bei der Erstellung einer Applikation kann man 'jedes Element' verwenden – HTML Elemente haben eine definierte Rolle -> ARIA überschreibt die eigentliche Rolle eines Elementes
– Assistive Technologien erkennen die 'Rolle' eines Elementes (und nicht das, was visuell dargestellt wird).
– Screenreader kann die Rolle dieses Elementes erkennen
– Wenn vorhanden, sollten Elemente mit inhärenter Semantik verwendet werden. In HTML etwa sollte <p> verwendet werden und nicht .
– Die Rolle eines Elements darf sich nicht zeitbasiert oder durch Aktionen der User ändern. Um die Rolle eines Elements zu ändern muss das Element samt aller Kindelemente gelöscht werden, und durch das gewünschte neue Element mit der entsprechenden Rolle eingefügt werden.

ARIA-Attribute (aria-attributname="value")

– Stellen der assistiven Technologie zusätzliche Informationen (Zustände und Eigenschaften) bereit.BSP: aria-valuemin/valuemax/valuenow/valuetext/labelledby

Document und Landmark-Rollen

– Helfen die Struktur eines Dokumentes näher zu beschreiben; bieten Sprungmarken innerhalb eines Dokumentes

Document-roles: article (Inhalt, der eigenständig Sinn ergibt, beispielsweise ein Blogeintrag), columnheader, definition, directory, document, group, heading, img, list, listitem, math, note, presentation, region, row, rowheader, separator, toolbar

Landmark-roles: application, banner (Site-spezifischer Inhalt, Titel, Logo), complementary (Unterstützender Inhalt für den Hauptinhalt, aber auch für sich alleinstehend wenn er vom Hauptinhalt getrennt wird), contentinfo (Fußnoten, Copyright-Hinweise, rechtliche Hinweise), form, main (zentraler Inhalt des Dokuments), navigation (zentraler Inhalt des Dokuments), search (Sektion mit Suchfunktion um die Website zu durchsuchen)

Verbesserung mit ARIA

<div id="menu" **role="navigation" aria-label="Hauptmenü"/**> - HTML
 - HTML
<input **aria-checked="false" role="checkbox"**/> - HTML
<p **id="desc1"**>Detailliertere Bildbeschreibung</p> - HTML
objCheck.setAttribute(**'aria-checked','false'**); /true /mixed
<nav role="navigation" **aria-label="Untermenü"**> - HTML5
<select name="color" required **aria-required="true"**> - HTML5

Werte (aria-attribute) werden mit JavaScript aktualisiert (im DOM) und können dann von assistiver Technologie ausgelesen werden.
Funktion bei ‚onclick' ! (Prüfe Tastaturbedienung) -> alt-Text auch mitändern

Live Regions

– Erlauben es Elementen mitzuteilen wenn sie verändert wurden, ohne dass der Nutzer den Fokus auf die aktuelle Aktivität verliert. **aria-live="value"; value = {off, polite (nachher), assertive (sofort)}**

aria-atomic="value" (optionale Eigenschaft für Live-Regionen, true oder false)

Wenn diese Region aktualisiert wird, dann wird dem Nutzer (=der assistiven Technologie) die gesamte (true) Region oder nur der geänderte Teil (false) mitgeteilt.

aria-relevant="value"

Ergänzend zu aria-atomic und aria-live. Gibt an, welche Art von Update bekanntgegeben wird - addition und removals (DOM Knoten), text -> Default ist text + additions

aria-busy="value"

optionale Eigenschaft; Wenn mehrere Bereiche geladen werden und die Eigenschaft ist true, dann wird gewartet bis alles fertig geladen ist. Dadurch wird verhindert, dass assistierende Technologien mit der Verarbeitung beginnen, bevor die Änderung abgeschlossen ist.

Einige Rollen haben 'automatisch' eine live Region. Können mit aria-live 'überschrieben' werden

alert: entspricht aria-live="assertive“
status: entspricht aria-live="polite"
timer: Enthält nummerische Angabe; Wird hoch/runter gezählt. Implizites aria-live="off"
 29
marquee: wenn sich (unwichtige)Information regelmäßig ändert (scrolling); implizites arialive="off" (sicherstellen, dass man das 'scrolling' auch ausschalten kann!)
log: wenn Text 'unten' angehängt wird und 'oben'verschwindet; implizites aria-live="polite"

Testtool für ARIA ->siehe Tools - Web-developer toolbar (FF) zeigt ARIA-Rollen an

User Agents & Authoring Tools

USER AGENTS

– **Ein User Agent ist jegliche Software, die WebSeiten oder Multimediainhalte darstellen kann**, Z.B. Web Browser, Media Player, 'Reader', Plugins
– Alles was 'Webinhalt rendert', lange Geschichte an Browsern, mit verschiedenen Zielgruppen
– Verschiedene Browser mit gleicher Renderengine
– Browser sind mehr als "reiner Anzeiger" ➔ viele Notwendigkeiten und Wünsche betreffend Barrierefreiheit
– Verschiedene Plattformhintergründe ➔ verschiedene (Basis) Zugänglichkeiten

AUTHORING TOOLS

Ein Authoring Tool ist jegliche Software, die Web-Seiten oder Multimediainhalte erstellen kann

• Lokale Software
– Software mit Speicher-/Exportfunktion (MS)
– Editoren/Website-Generatoren (Dreamweaver)

• Online Software
– Konverter (Content Management Systeme)
– Editoren

ZUGÄNGLICHKEITSBEWERTUNG

Accessibility Guidelines

(Web Content) WCAG für den Content

(User Agent) UAAG für "die Wiedergabe"

(Authoring Tool) ATAG für "das Erstellen"

(Programmierer auch bei UAAG, ATAG)

UAAG

– Working Group Note 15 December 2015 (complete)
"W3C does not currently plan to advance UAAG 2.0 to Recommendation status" (würde einen zu großen formalen Testaufwand bedeuten)
– Ist mit WCAG 2.0 und ATAG 2.0 als Gesamtkonzept zu sehen
– Unterteilt in 5 Prinzipien und 26 Richtlinien
– Bewertet mit A, AA und AAA
trifft zu für z.B. Stand-Alone Browser, Embedded Browser, Editoren die Web Inhalte anzeigen können (z.B. MS Word), Authoring Tools die Web Inhalte anzeigen können, Webbasierte UA (z.B. Google Translate), Media Player (z.B. Realplayer), Browser mit eingeschränktem Umfang (z.B. mobile Apps)
aber NICHT zu für z.B. Einfache Texteditoren (z.B. Notepad), Betriebssysteme (auch wenn sie HTML verwenden)

UAAG 2.0 – 5 Prinzipien

– Wahrnehmbar Ensure that the user interface and rendered content are perceivable

Alternativer Inhalt

Der Anwender kann wählen ob alternativer Inhalt verwendet werden soll und welcher Inhalt dauerhaft gezeigt werden soll. Es wird empfohlen eine Anzeigekaskade definieren zu können. Steuerelemente dürfen dabei nicht leiden. Größe und Position von alternativen Inhalten sollen einstellbar sein.

Fehlender Inhalt

Der Anwender kann fehlenden alternativen Text anfordern. Der Anwender kann den UA um einen automatischen ‚bestguess' bitten.

Hervorhebungen

Der Anwender kann visuell zwischen verschiedenen Link. Typen (besucht, fokusiert, ...) unterscheiden. Kann 'highlighting optionen' wählen (VordergundHintergrund, Bordercolor, Dicke des Rahmens). Der Anwender kann die visuelle Markierung individuell einstellen (Schriftschnitt, Schriftgröße, Rahmen, ...).

Text Konfiguration

Der Anwender kann die Textanzeige beeinflussen (Schriftgröße, Farben, Stil, ...). Dabei auch Autorangaben überschreiben! Der Anwender kann optional jeden Text in der gleichen Größe haben.

Lautstärke Konfiguration

Der Anwender kann jede Audiospur relativ zur Hauptlautstärke einstellen.

Künstliche Stimm-Konfiguration

Wenn künstliche Stimmen verwendet werden, können die üblichen Parameter (Geschwindigkeit, Lautstärke, Stimme, ...) vom Anwender gewählt werden. Zusätzlich sollen spezielle Parameter wie Betonung, Buchstabieren u.ä. einstellbar sein.

Anwender Style Sheets

User Style Sheets sollen einstellbar sein. Die Wahl zwischen Anwender Style Sheet und Autor. Style Sheet soll möglich sein. Anwender Style Sheets sollen speicherbar sein.

Orientierung in Viewports und Fenstern und Einstellungen von Viewports und Fenster

Es soll erkennbar sein was aktiv ist. Der Fokus (visuell) darf nicht verloren werden. Der Viewport soll größenveränderbar sein. Scrollbalken sollen den sichtbaren Bereich verschiebbar machen und anzeigen wo man sich befindet. Der graphische Inhalt soll bei Zoom mitvergrößert werden. Der Fokus und das Karret soll bei 'Resizing' oder Formatänderungen sichtbar bleiben. Das automatische Öffnen von Fenstern oder Tabs soll konfigurierbar sein. Der Fokus soll automatisch auf neue Fenster und Tabs wechseln. Der Viewport soll Scrollbars haben/erlauben (wenn notwendig). Alle Fenster sollen optional die gleichen. Bedienelemente haben. Sprungmarken sollen setzbar sein, auf die man zu einem späteren Zeitpunkt mit Kurztasten zurückkehren kann. Fixe Einheiten können vom User überschrieben werden. Inhalt kann linearisiert werden.

Alternative Ansichten

Der Anwender kann optional den Source sehen. Der Anwender kann optional ein/e Übersicht/Inhaltsverzeichnis sehen.

Element Information

Der UA kann optional Zusatzinformationen zu Elementen anzeigen (Summary, Headers, ...). Der UA kann optional Zusatzinformationen zu. Links anzeigen (intern/extern, Titel, ...).

– Bedienbar Ensure that the user interface is operable

‘Modality independent' - Alles soll mit der Tastatur bedienbar sein:

Jede Funktion, Das Bewegen des Fokus, Tastaturfallen vermieden, Typische Tastenkombinationen der Plattform müssen unterstützt werden.

Sequentielle Navigation muss ermöglicht werden:

Mit dem Keyboard kann man sich sequentiell durch den Inhalt bewegen, Fenster/Tabs sequentiell mit dem Keyboard durchgehen, Die sequentielle Navigation kann optional nicht von vorne beginnen.

Direkte Navigation und Aktivierung soll unterstützt werden:

Direkt zu wichtigen Elementen (Links, Strukturelementen, ...) navigieren, Optional kann der Anwender aktivierbare Elemente direkt anspringen und aktivieren. Tastenabkürzungen können optional bei den Elementen angezeigt werden. Der Anwender kann die Tastenabkürzungen ändern und speichern.

Textsuche soll verfügbar sein:

Text soll durchsuchbar sein (inkl. alternativer Inhalte). Text soll in verkehrter Richtung durchsuchbar sein. Treffer der Suche sollen in der Ansicht markiert werden. Der Anwender soll informiert werden, wenn kein Treffer erzielt werden konnte. Alternativer Inhalt soll durchsuchbar sein.

Strukturierte Navigation soll unterstützt werden:

Navigation zumindest folgendermaßen möglich: Überschriften, Abschnitte bei Inhalten, Innerhalb von Tabellen.

Bevorzugte Einstellungen:

Bevorzugte Einstellungen können zwischen Sessions gespeichert werden. Es gibt 'default settings'. Man kann verschiedene Gruppen von bevorzugten Einstellungen haben. Bevorzugte Einstellungen können zwischen Geräten portiert werden.

Grafische Steuerelemente anpassen:

Benutzer können einstellen, welche Steuerelemente, Funktionen, Addons, dargestellt werden: Anzeigen, Vereinfachen (z.b. Verstecken), Repositionieren, Zurücksetzten (auf default-Werte).

Zeitunabhängige Interaktion:

Es gibt keine Zeitlimits oder es existieren Einstellmöglichkeiten: Ausschalten, Einstellen (bis zum 10-fachen), Ausnahmen in wichtigen Fällen.

Der UA unterstützt den Anwender zeitbasierte Medien zu steuern:

User können Laden und Abspielen von zeitbasierten Medien steuern. Zeitgesteuerte Medien können durch Platzhalter ersetzt werden. Das Abspielen von Medien kann verändert werden (Abspielart, Helligkeit, Lautstärke, Geschwindigkeit...).

Andere Eingabegeräte sollen unterstützt werden:

Wenn eine Eingabegerät von der Plattform unterstützt wird, dann kann jede Funktionalität des UA inklusive Texteingabe von diesem Eingabegerät bedient werden.

Der UA soll den Anwender unterstützen Fehler zu vermeiden und/oder zu korrigieren:

'Back Button', Undo Funktionalität, Sicherheitsabfrage vor dem Absenden von Forms, Formdaten merken, Rechtschreibkontrolle

– Verständlich Ensure that the user interface is understandable

Der UA soll (inkl. der ZugänglichkeitsFunktionen) verständlich dokumentiert sein – **Zugängliche Dokumentation** (!)
Der UA soll sich vorhersehbar verhalten – Der Fokus soll nicht unvorhersehbar wechseln

– Programmzugang (API) Facilitate programmatic access

Die Zugänglichkeitsfunktionen sollen über eine API ansprechbar sein

Unterstützung von plattformspezifischen Schnittstellen. Eigenschaften und Werte von UI Komponenten (inkl. gerenderter Inhalt) sind zur Verfügung gestellt: Name, Rolle, Status, Selektiert, Focus, Fontfamily, Vordergrund- und Hintergrundfarbe von Text, Tastaturkommandos, Carret Position...

Zugang zum DOM soll gegeben sein. Zugang zu Interaktionselementen soll gegeben sein: Checkboxen, Text Edit Area.

– Spezifikationen & Konventionen Comply with applicable specifications and conventions

Der UA soll den plattformüblichen Spezifikationen und Konventionen genügen:
Wird z.B. im UA selbst HTML verwendet, so sind diese WCAG konform zu gestalten. Implementiere Accessibility features der Plattform. Es soll möglich sein, Inhalt in alternativen Viewern darzustellen.

ATAG

= **Jegliche webbasierte oder nicht webbasierte Software, die von Autoren (alleine oder kollaborativ) verwendet werden kann, um Inhalte für das Web (für andere Anwender oder Autoren) zu erstellen oder zu ändern.**
BSP: Web Seiten Tools, Integrierte Web Development Tools, Blogs, Wikis, Multimedia Authoring Tools
NICHT einstellbare persönliche Portale, Bestellformulare, Standalone Accessibility Checker

Unterteilung in 2 Teile
- **Je Teil 4 Prinzipien**
- **Jedes Prinzip in Richtlinien unterteilt (13 + 11)**
- **Konformitätslevel A, AA und AAA (siehe WCAG2)**

– Part A: Das AT User Interface zugänglich gestalten

AT User Interfaces sollen den Zugänglichkeitsrichtlinien entsprechend:

Webbasierte Funktionalität ist zugänglich (konform entsprechend WCAG2). Nicht Webbasierte Funktionalität ist zugänglich entsprechend existierender Plattform-Richtlinien.

Edit-Ansichten sollen wahrnehmbar sein:

Alternativer Inhalt wird den Autoren zur Verfügung gestellt. Auf die zu editierenden Inhalte und den Status soll man programmtechnisch zugreifen können (Statusindikatoren, Text-Properties).
Edit-Ansichten sollen bedienbar sein.

Alle Funktionen des AT sind mit der Tastatur bedienbar (Keine Tastaturfallen, Effiziente Tastaturbedienung...):

Autoren haben genug Zeit das UI zu bedienen (Ausschalten, Einstellen von Limits...). AT helfen 'Blitze' zu vermeiden. Navigation und Editieren durch Strukturierung vereinfachen. Textsuche für den Content. Verwaltung von 'preference settings': Einstellungen der Anzeige des AT sollen keine Auswirkungen auf den generierten Content haben; Einstellungen sollen während des Editierens von Inhalten verändert werden können; Änderungen der Plattform werden vom AT übernommen (außer es ist etwas anderes eingestellt). 'Vorschauen' sind zumindest so zugänglich wie marktübliche Useragents.

Edit-Ansichten sollen verständlich sein:

Autoren werden unterstützt Fehler zu vermeiden und auszubessern. Das UI soll (inkl. der Zugänglichkeits-Funktionen) verständlich dokumentiert sein (Zugängliche Dokumentation).

– Part B: Das AT soll Autoren bei der Erzeugung von zugänglichen Inhalten unterstützten

Vollautomatische Prozesse sollen zugänglichen Inhalt erstellen:

Automatisch generierter Inhalt soll zugänglich sein (entsprechend WCAG 2). Zugänglichkeitsinformationen sollen erhalten bleiben, bei Restrukturierung, Recodierung, Copy-Paste, Optimierungen.

Autoren sollen beim Erstellen von zugänglichen Inhalten unterstützt werden:

Es soll sichergestellt werden, dass zugänglicher Inhalt erstellt werden kann, z.B. Alt-Text bei Bildern kann editiert werden. Autoren sollen bei der Erstellung zugänglicher Inhalte angeleitet werden z.B. zugängliche Methoden werden angezeigt (und bevorzugt). Entsprechende Settings werden bevorzugt angegeben.

Autoren sollen beim Managen von alternativen Inhalten unterstützt werden:

Beim Hinzufügen von Nichttext-Inhalten sollen die alternativen Inhalte programmtechnisch geändert werden können. Alternative Inhalte werden mit der Quelle für eine spätere Verwendung gesichert und können im Nachhinein geändert und gelöscht werden.

Autoren sollen mit zugänglichen Vorlagen unterstützt werden:

Die Zugänglichkeitsstufe wird bei Vorlagen angezeigt. Die der Zugänglichkeit widersprechenden Unterschiede einer Vorlage werden angezeigt.

Autoren sollen mit zugänglichen Inhalten (preauthored content) unterstützt werden:

Die Zugänglichkeit von vorgegebenem Inhalten soll bekannt sein und soll angezeigt werden.

Autoren sollen unterstützt werden, die Zugänglichkeit von Inhalten zu verbessern:

Unterstützung zur Accessibility Überprüfung soll angeboten werden (entsprechend WCAG 2). Autoren sollen unterstützt werden die Zugänglichkeit herzustellen. Vorschläge zu Verbesserungen werden bei erkannten Fehlern angezeigt.

AT sollen Zugänglichkeits-Funktionen integrieren und bewerben:

Zugänglichkeits-Funktionen sollen verfügbar sein. Zugänglichkeits-Funktionen sollen per default aktiviert sein. Alles was man ausschalten kann, soll man auch wieder einschalten können. Wenn man Zugänglichkeits-Funktionen ausschaltet, soll erklärt werden, welche Probleme dadurch entstehen können. Alle zugänglichen Features sind zumindest so leicht zu finden wie nicht zugängliche Features.

Die Dokumentation soll Zugänglichkeits Funktionen bewerben:

Beispiele. Beschreibung der Accessibility Support Features. Enthält ein Tutorial. Eigener Index der Zugänglichkeits-Funktionen.

Check-Tools und deren Verwendung

Verschiedene Tests

Es gibt (bisher) kein Tool, das sicher und ohne menschliche Hilfe eine Seite vollständig prüfen kann. Es gibt kaum so viel Zeit, dass ein Mensch eine Seite ohne ‚maschinelle' Unterstützung testen kann. → Testdurchführung von einem geübten Menschen mit Unterstützung von Tools.

Es ist Empfehlenswert verschiedene Tools selbst auszuprobieren und die für sich besten selbst zu finden. → Use with care!

(X)HTML Konformität

Vollständig ‚maschinell' durchführbar. Prüft gegen eine gegebene DTD (Doc Type Definition). Daher verschiedene Angebote für **online** (überall aber langsam und privacy issues) - W3C, Tidy, Wave, HTML5
und **offline** Tests (in jedem HTML/normal Editor enthalten, standalone) - HTML Tidy, CSE Validator, HTML Validator Plugin (FF)

CSS Konformität

online: W3C, CSS Portal, TopStyle, Web-Developer, Dreamweaver, Entwicklertools

Browsertests

Browserunabhängigkeitsprobleme bei Verwenden von: neueren Standards, Skripts, beliebten aber nicht genormten Eigenschaften, Komplizierten CSS, Wunsch nach pixelgenauer Ausrichtung.
Bei verschiedenen **Herstellern** i.a. leicht → einfach mehrere Browser installieren.
Bei verschiedenen **Versionen** eines Herstellers: Bei manchen Browsern ist eine Parallelinstallation möglich ODER Verschiedene virtuelle Maschinen mit verschiedenen Versionen verwenden.
BSP: Browsershots, IETester, TopStyle

WAI Checkers Zugänglichkeit(WCAG2.0)

Nur teilweise maschinell überprüft weil Ergebnissen aus rein maschinellen Tests nicht 100% trauen aber i.A. eine gute Grundlage. Zusatzüberprüfungen sind notwendig. Z.B.

Farbgebung, erklärende Texte, Einfache Sprache, Inhalte, Anpassung an den Browser, Fenstergröße
BSP: TotalValidator, CSE HTML Validator
Weitere Checktools: Farbhelfer (Color Contrast, Color Blind)

Browserintegrationstests

Keiner der ‚großen Browser' hat echte Checkfeatures integriert (Begonnen haben IE). Firefox hat mit einem sehr flexiblen PluginSystem großes Potential als Basisplattform ➔ Viele für Web-Developer interessante Plugins. Chrome hat das Pluginsystem übernommen➔daher auch viele Plugins für Chrome verfügbar.
BSP: HTML Validator (offline), Web-Developer (FF- online), TotalValidator, Popup ALT Attribute, WAVE

Hilfstools

Berichte
Abhängig vom Ziel (Gutachten über Erfolg,Feedback an Entwickler, ...), e.g. Fireshot
Erhebungsprotokolle
WAT (Online) (Web Accessibility Tickler)

Cascading Style Sheets-JavaScript

CSS3

CSS Snapshot 2020 - This document collects together into one definition all the specs that together form the current state of Cascading Style Sheets (CSS) as of 2020. The primary audience is CSS implementers, not CSS authors, as this definition includes modules by specification stability, not Web browser adoption rate.

CSS is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on paper, in speech, etc.

Es gibt keine Versionsangabe wie bei html. Der Grundaufbau ist:
Selektor {Eigenschaft1: Wert; Eigenschaft2: Wert; ...}
Mehrfaches Vorkommen / Überschreiben es gilt immer der zuletzt zugewiesene Wert.
Unbekannte Eigenschaften oder Werte werden ignoriert.

Herstellerspezifische Eigenschaften (in früheren Versionen von CSS3/Browsern)
-moz (Gecko-basierte Browser FF, Seamonkey)
-webkit (Webkit-basierte Browser Safari, Chrome, Konqueror)
-ms (IE)
-o (Opera)

Unterschiedliches 'Aussehen' für verschiedene Medien -> Media Queries
Damit ist "Responsive Design" möglich - ! Menü-Darstellung angepasst für unterschiedliche Geräte

Bei *externen Stylesheets*:
<link rel="stylesheet" href="bildschirm-stylesheet.css" **media="screen/print"**>

Innerhalb eines *Stylesheets*
@media print {/* zusätzliche oder abweichende CSS-Regeln fürDrucker */}

Mit Media Queries sind auch Eigenschaften von Geräten abfragbar:
@media (max-width: 915px)

Auch im <head> des Browsers für richtiges Verhalten.
Testen Responsive Design: Web Developer Toolbar FF, Plugin Chrome, minimise Browser

Schrift-/Textformatierung

- Alle Arten der Schriftart, -Größe, -Stil, -Farbe
- Zeichen-/Wortabstände, Schatten, Outlines, Zeilenumbruch
- Ausrichtung von Text (horizontal & vertikal), Einzüge, Zeilenumbrüche
- Relative Schriftgrößen (em = 100%, ex, ch, %, rem)
- Absolute Einheiten (cm, nm, in, px, pt, pc) - vermeiden!
- Wichtig für Zooming als zweite Möglichkeit zum 'Browserzoom'
- Um user-stylesheets (besser) zu bedienen
- -> Schriftgrößen vom 'Elternelement' geerbt

Pseudoklassen und -elemente

Pseudoklassen sind 'Key-Words' die nach einem Selektor stehen und ein Element ansprechen oder einen Teil des angesprochenen Elements ändern, wenn es einen gewissen Zustand hat.
Bei CSS2 mit : geschrieben, ab CSS3 mit :: (first-line, first-letter, before, after, selection)
Strukturelle (:first-child, :last-child , :nthchild(odd, even, an+b))
Dynamische (

- :link, :visited
- :hover, :active, :focus
- :target
- :enabled, :disabled, :checked
- :valid, :invalid)

Gestaltungselemente

Rahmen, Schatten & Outlines beliebiger Elemente. Transformationen 2D und 3D.

Hintergründe und Bilder

Hintergrundbilder und deren Größe, Wiederholung; Hintergrundfarben; Hintergrundgradienten

JS

JavaScript erlaubt verbesserte(?) Interaktion, Informationsverarbeitung, Bearbeitung von Webcontent. JavaScript kann (muss aber nicht) Probleme mit der Accessibility machen.

- **Laut WCAG 1 muss man eine Seite auch ohne JavaScript bedienen können.**
- **Laut WCAG 2 darf JavaScript gefordert werden, es muss aber zugänglich sein**

Mögliche Probleme
Navigation: Es kann sein, dass man die Webseite nicht mehrmit der Tastatur oder assistierender Technologie benutzen kann.
Versteckter Inhalt: Inhalt oder Funktionalität einer Webseite werden so angeboten, dass sie für assistierende Technologien nicht mehr zugänglich sind.
Kontrolle durch den Benutzer. Benutzer können Content der sich automatisch ändert nicht mehr kontrollieren.
Verwirrendes Verhalten: Die ‘normale' Funktionalität des User-Agenten kann verändert werden.
Event handler (‘Ereignisbehandler’; ‘Script-Teile’) werden vom Browser oder Benutzer ‘getriggert’, d.h. ausgelöst. z.B.:
– wenn eine Seite geladen wurde
– wenn ein Mausklick stattgefunden hat
– wenn eine gewisse Zeit erreicht wurde.

- **Device dependent** (geräteabhängige) Eventhandler: MausEvent, Tastatur-Event
onMouseOver und onMouseOut
onKeyDown und onKeyUp nur für die Tastatur
onClick (+onDblClick) (dabei wird auch ENTER von Browsern erkannt) - Funktioniert aber nicht bei klickbarem Text, Tabellenzellen, usw. ->‘ENTER’ und ‘SPACE’ muss im Script gesondert abgefragt werden.

- ***Device independent*** (geräteunabhängige) Eventhandler: Wenn ein Event von Maus und Tastatur ausgelöst wird, (oder andere Ereignisse die nicht vom Eingabegerät abhängen...)
onFocus und onBlur
onChange und onSelect - Ergebnisse dieser Events müssen natürlich accessible sein.

Um Zugänglichkeit zu gewährleisten muss man also entweder geräteunabhängige Eventhandler verwenden, oder sowohl Maus als auch Tastatur entsprechen abfragen.

Dynamischer Inhalt

Mit **document.write** kann Inhalt generiert werden. Dieser Inhalt muss zugänglich bleiben (schnelle Änderungen, Focusänderungen...). Änderungen bekanntgeben (ARIA).

Wenn scripting nicht zugänglich gemacht werden kann -> Serverside processing. (z.B. Formularkontrolle)

<noscript> (sollte heutzutage nicht mehr vorkommen!)

• Wenn Javascript deaktiviert ist->noscript Inhalt.
• Wenn JavaScript enabled ist verarbeiten assistierende Technologien (screenreader) das script -> noscript ist kein Weg um schlechtes scripting zu ersetzen!

Hilfsmittel - Assistive Technology

Hilfsmittel für technikferne Menschen

– Technikfern keine Behinderung im direkten Sinn
– Oft im Zusammenhang mit anderen Behinderungen (z.B. bei alten Menschen) wie Fehlsichtigkeit, Hörbehinderung, motorische Einschränkungen, ...
– Kaum echte Hilfsmittel, aber viel im Design zu beachten

Webdesign

– Begriffe immer erklären und nicht voraussetzen

– Keine Bedienkonzepte eines Betriebssystems voraussetzen

– In den Hilfen/Erklärungen auch simple Dinge erläutern und nicht voraussetzen

– Symbole können ohne Vorkenntnisse falsch oder völlig anders verstanden werden

– Bedienelemente eindeutig aussehen lassen (was ist ein Link, was ist ein Button, wo kann ich etwas eintragen, ...)

– Multimodale Ausgaben vorsehen

– Text und Symbole kombinieren

– Symbole sollten in Form und Farbe verschieden sein

– Darauf achten, daß man auch ohne (viel) Scrollen die Bedienungselemente sieht

– Mehrfache Scrollbalken vermeiden (z.B. Frame und Browser)

– Seiten einfach gestalten und nicht überfüllen

– Sicherheitsabfragen bei Aktionen einbauen

– Klare, nicht technische Formulierungen verwenden

Hörbeeinträchtigungen

– Korrektur mit üblichen Hörhilfen ➔ (i.a) nicht webrelevant

– Starke Hörbeeinträchtigungen ➔ Probleme mit manchen Webdiensten möglich (z.B. Sprachtelefonie, Hörbücher, ...) bzw. Hintergrundmusik/-geräusche störend

– Windows Bordmittel

Gehörlose Menschen

– Einschränkungen bei Informationsaufnahme wie Signalgeräuschen, Videos, Hintergrundmusik, ... ➔ Ersatz notwendig

– Bedarf bzw. Bevorzugung von Gebärdenvideos ➔ Zusätzlich zu Texten Gebärdenvideos anbieten

– Gehörlose Menschen ‚sprechen' Gebärdensprache. Deutsch oder Englisch sind für sie Zweit- oder Drittsprachen.

Webdesign

– Videoübertragung (für Gespräche) muß schnell genug für Gebärdensprache sein
– Musik oder andere Geräusche sollten abschaltbar sein
– Keine Information, die nur im Audio ‚verborgen' ist
– Videos mit Untertiteln versehen
– Zusätzlich zum Text Videos in Gebärdensprache anbieten

Körperliche Einschränkungen

– Untere vs. Obere Extremitäten
– Sehr abhängig, wie stark die Einschränkungen sind
– Verschiedene Kategorien
    - Unterstützungen bei Eingabegeräten
    - Ersatz von Eingabegeräten
    - Beschleunigende Hilfsmittel

Mechanische:

- Unterstützung mittels Mechanische Arm- Beinunterstützung / Gerätehalterungen, Lochmasken
- Schutzhüllen
- Alternativen via Touchscreen, Trackball, Keyboard, Tasten, Joystick, Special-Mouse (Integramaus, FlipMouse, Tastenmouse, Keymouse, Handmaid), Klein- und Großtastaturen (mit Fingerführung), Einhandtastaturen (Tipy), Keyboards mit Overlays, Eye Tracking
- Software Support via Windows Keyboard Settings, EMU for faster writing, Onscreen Keyboard, Spracheingabe Einstellungen, Scanning Systems (ISUI)

Webdesign

– Maus darf keine Voraussetzung sein
– Die Eingabegeschwindigkeit muß auch sehr langsam sein können
– Grobmotorische Bewegungen dürfen kein unüberwindbares Hindernis sein
– Plugins (Flash & Co) können mit Hilfswerkzeugen große Probleme verursachen

Sehbeeinträchtigung

Sehbeeinträchtigungen, die mit einer Brille korrigiert werden können → nicht besonders web relevant. Vergrößerungsprogramme (in Windows integriert oder als Zusatzprodukt), z.B. Bildschrimluppe und Farb-, Mauszeiger-, Carret und Kontrastanpassungen in

Windows, Zusätzliche Soundunterstützung (Sound Options), Browsereinstellungen: Schriftgröße/-art, Eigene Farbschemata, Custom-CSS (Stylish-plugin, Greasemonkey)

Webdesign
– Der Sichtausschnitt kann durch Vergrößerung sehr eingeschränkt sein
– Volle und große Seiten sind sehr kompliziert zu betrachten
– Schriftgrößen/-arten und Farben können überschrieben werden (CSS kann überschrieben oder ausgeschaltet werden)
– Gute Kontraste sind extrem wichtig
– Eine gute Führung bei Vorgängen ist wichtig
– Der Mauszeiger kann völlig anders aussehen
– Es können z.T. CSS Eigenschaften überschrieben werden
– CSS könnte gänzlich deaktiviert sein
– Zusätzliche Sounds können fehlende Übersicht ersetzen
– Verschiedene Zoom-Funktionen können problematisch für Designs sein

Blinde Menschen habe andere Bedürfnisse als sehbehinderte/fehlsichtige Menschen.

– Screenreader: einen für Sehende ungewöhnlichen Zugang zu Web-Seiten
– Selfvoicing Browsers
– Braillezeilen
– Brailletastaturen

Webdesign
– Jede Seite muß gut strukturiert sein (Überschriften, Listen, Links, ...) und nicht nur strukturiert aussehen
– Grafiken brauchen einen sinnvollen Alt-Text und im Bedarfsfall eine Description
– Links brauchen einen sinnvollen Text (auch ohne Textzusammenhang)
– Sounds können optische Hinweise ersetzen
– Optische Captcha brauchen Soundalternativen
– Die Seite muß vollständig mit der Tastatur alleine verwendbar sein
– Hintergrundmusik oder nebenbei laufende Filme stören das Sprachverständnis für den Screenreader
– Tabellen sind aufgrund der graphischen Darstellung schwierig zu lesen
– Filme ohne Untertitel oder erklärenden Text sind i.A. nicht nur nicht zugänglich sondern störend
– Seiten sollen möglichst ‚klein' gehalten werden

– Accesskeys/links sind u.U. sehr hilfreich

HTML(5)

HTML

General

1989/90 erste HTML Version Gemeinsam mit neuem Protokoll HTTP → World Wide Web
1992 Mosaic Browser (erster graphischer Browser)
1994 World Wide Web Consortium (W3C) - ein internationales Konsortium in dem Mitgliedsorganisationen, fix angestelltes Team, und Öffentlichkeit Web-Standards entwickeln
– Finanziert sich durch Mitgliedsbeiträge, öffentliche und private Finanzierung
– Geleitet vom USA, Europa und Japan
– World Offices in mehreren Regionen (lokale Arbeiten)
1995 RFC 1866 als HTML 2.0 formal definiert
1995 Microsoft veröffentlicht ersten Internet Explorer • -> „Browserkrieg”
->Spracherweiterung der Browserhersteller bewirkt, dass HTML seine Universalität verliert
Versuch zu standardisieren: HTML+ ; HTML 3.0
1997 HTML 3.2 vom W3-Consortium (W3C) als neuer (Kompromiss) Standard definiert.
1998 HTML 4.0 ; weist auf konsequente Trennung von Inhalt und Präsentation hin; Style-sheets (CSS)
1999 leichte Überarbeitung: HTML 4.01
2000 Neuformulierung von HTML 4.01 in XML 1.0 -> XHTML 1.0 (The Extensible Hyper Text Markup Language)
2006 W3C veröffentlicht Entwurf von XHTML 2.0 (Working Draft); Sprache soll abwärtskompatibel bleiben und in 95% der verbreiteten Webbrowser funktionieren
2010 W3C stellt Arbeit an XHTML 2.0 ein
2010 Arbeit an HTML 5.0 (bisher parallel zu XHTML 2) wird intensiviert

(X)HTML

HTML basiert auf SGML (Standard Generalized Markup Language), sehr ausgereift, gilt aber gemeinhin als ziemlich „sophisticated“.
Reduzierte Variante (sub-set) XML (Extensible Markup Language)

Wunsch HTML auf Basis XML Definition aufzubauen -> XHTML
Daher: **XHTML ist (sehr) ähnlich HTML (4.01)**
XHTML ist 100% kompatibel mit andere XML Sprachen (Scalable Vector Graphics SVG, Wireless Markup Language WML, Synchronized Multimedia Integration Language SMIL...)
XHTML ist „erweiterbar“, neue „Elemente“ können einfach hinzugefügt werden
Unterschied zu HTML erst merkbar, wenn Code nicht von Browsern sondern von „anderen Programmen“ verarbeitet wird

HTML = Hyper Text Markup Language
HTML ist eine statische Sprache (man kann also nicht ‚programmieren‘), formale Sprache (so wie Programmiersprachen, C, Pascal, Perl etc.)
HTML ist eine Auszeichnungssprache(Markup) d.h. Tags werden verwendet um den Inhalt eines Dokuments zu beschreiben.
Ein HTML-dokument besteht aus tags und plain Text. <tag>plain Text</tag>
Tags können zusätzliche Angaben enthalten -> **Attribute** (Attribute haben Werte)
<tag attribut=“wert“ >Text</tag>

XHTML Dokumente müssen „wohlgeformt“ sein:
Jede Element muss semantisch markiert sein.
Jedes Element muss ein End-Tag haben (auch ein leeres Element!)
Tags müssen klein geschrieben werden
Tags müssen korrekt geschachtelt werden
Attributwerte müssen immer in " " (doppelte Anführungszeichen) geschrieben werden
Kurzschreibweise von Attributen ist nicht unterstützt
Es gibt Pflichtattribute die immer angegeben werden müssen
Sprungmarken mit id="" und nicht name=""
Struktur eines (X)HTML-Dokumentes
1) Doctype declaration -> Muss am Anfang stehen

HTML Inhalt:
2) HTML-Kopf (HEAD) wird üblicher Weise nicht dargestellt. Diverse Informationen für und über das Dokument
3) HTML-Körper (BODY) Informationen die im allgemeinen im Browser angezeigt werden
Zeichenkodierung empfohlen <?xml version="1.0" encoding="UTF-8" ?>
sonst kann der Browser entweder UTF-8 oder UTF-16 verwenden.

Doctype – Nimmt Bezug auf DTD (Dokumenttyp-Definition) und Sprachversion der Datei
Wird von Browsern (und andere Programmen) verwendet um die Seite darzustellen (kann die Darstellung bei Syntaxfehlern verhindern)
DTD (Dokumenttyp-Definition)
– Zu Beginn eines XML Dokumentes (vor dem Wurzelelement); Verweis auf eine Datei (URI) oder inline
– Für standardisierte DTD Verweis auf Public Identifier
– Gibt an welche Elemente ein Dokument enthalten darf
– Welche Elemente innerhalb von welchen vorkommen dürfen
– Attribute die zu einem Element gehören und ob die Angabe Pflicht ist oder nicht…

Doctype – Varianten:
• ‚Frameset‘: Wenn Frames verwendet werden
• ‚Transitional‘: Vergibt kleine Fehler– (erlaubt Verwendung von ‚depreciated‘ Elementen und Attributen z.b. <font>, bgcolor, align)
• ‚Strict‘: Volle Standardkonformität gefordert
Ohne (oder mit falschem) Doctype -> ‚Quirks‘ Mode

Hierarchische Struktur
• Elemente müssen korrekt geschachtelt werden
• Insgesamt 80++ Tags
• Davon im <head> sechs verschiedene Elemente: <title>, <base>, <link>, <script>, <meta>, <style>

XHTML Dokument
– XHTML verlangt eigentlich “drakonische Fehlerbehandlung“
– Beim ersten Auftreten eines Fehlers Abbruch. Wird mit MIME-Typ application/xhtml+xml erzwungen.
– Wird aber so gut wie nie gemacht, es werden XHTML-Dokumente (fast immer) mit dem MIMEType text/html ausgeliefert. (kleiner Widerspruch zur Programmierung in XHTML)

Document Object Model (DOM)
– ist eine Spezifikation für den Zugriff auf beliebige Elemente einer Auszeichnungssprache. (Schnittstelle)
– Also auch (aber nicht nur) für den Zugriff auf HTML sondern auch XML-Dokumente.
– Die Spezifikation wird vom World Wide Web Consortium (W3C) definiert.
– Ist keine eigene Programmiersprache sondern definiert Objekte, Eigenschaften und

Methoden die eine Programmiersprache umsetzten soll um DOM-fähig zu sein
– erlaubt Programmen den Inhalt, die Struktur und das Layout eines Dokuments dynamisch zu verändern.
– JavaScrip ist die Anwendung schlechthin die mit DOM arbeitet
– DOM ist nicht nur „Dynamic HTML“, es ist nicht nur auf die Client-Seite (Browser) beschränkt. Auch serverseitig kann es eingesetzt werden (z.B. CGI) um dynamisch Dokumente zu erzeugen.
• Eine Auszeichnungssprache (HTML, XML) kann als hierarchische Baumstruktur dargestellt werden
• Einzelne Bestandteile sind Knoten (node)
• Wichtigste Knoten:
  – Dokumentknoten - stellt gesamte Baumstruktur dar
  – Elementknoten - entspricht einem Element in HTML
  – Attributknoten - entspricht einem Attribut in HTML
  – Textknoten - stellt den textuellen Inhalt eines Elementes
• Knoten haben eine ‚Eltern-Kind‘ Beziehung
• Ausnahme sind Attributknoten, sie sind Eigenschaften von Elementknoten
• DOM-Baum Must-know-list: Überschrift, Paragraph, Liste, Div, Tabelle, Bild, Objekt

Tags
<**body**>
“Alles was dargestellt wird, ist da drinnen“
Block- und/oder Inline Elemente
  – Block-Elemente sind wie ein Container für ihren Inhalt
  – Inline Elemente unterbrechen den Textfluss nicht.
Mit CSS kann diese Verhalten aber umdefiniert werden
• Das <**p**> tag: Text muss ‚zumindest‘ in einen Paragraph <p> geschrieben werden
• Markup innerhalb eines Textes
– <**em**> – Emphasize, Hervorheben
– <**strong**> - Stronger Emphasize, starkes Hervorheben
Wird per default meist kursiv bzw. fett dargestellt. Daher auch oft fälschlich als eine andere Art von **<i>**(italic) und <**b**>(bold) gesehen. <b> und <i> sind Style-Definitionen und dürfen(sollen) daher nicht im HTML Code verwendet werden. Verwende CSS für die Formatierung!
• **Markup** im Text <blockquote>, <sup>, <sub>, <cite>...
Zitat (Einrückung), superscript , subscript, Zitation in html4, Titel eines Werkes in HTML5

**
** – Harter Zeilenumbruch. Die einzige Möglichkeit einen Zeilenumbruch fix zu definieren. Der Zeilenumbruch im HTML Code wird ansonsten ignoriert. Das ist wesentliches Eigenschaft um ‚flexibles Layout' gestalten zu können.

• Überschriften sind wesentliches Strukturelement

– Werden mit dem <**h_**> (heading) tag definiert

– Überschrift erster Ordnung : <h1>Überschrift</h1> (heading 1)

– Gliederung bis h6 möglich

– Formatierung im Stylesheet

– Nur für 'Überschrift' verwenden

• Listen sind ebenfalls wesentliches Strukturelement

– <**ol**> Ordered list - Nummerierte (sortierte) Liste

– <**ul**> Unordered list - Unsortierte (nicht nummerierte) Liste

– Bei beiden wird <li> für die Listenelemente verwendet

• Description list – <dl> <dt> <dd> term, description

• Listen zur Navigation

– **uo** Lists eignen sich sehr gut zum Aubau von Menüs

– Durch entsprechenden Gestaltung im Style-sheet können auch horizontale Menüs leicht gestaltet werden-> sind auch für Assistierende Technologien gut zugänglich

• Das ** ** tag: definiert eine 'division', d.h. einen 'Bereich' innerhalb des HTML Dokumentes.

– Wird verwendet um andere Elemente zu gruppieren

– i.A wird das im Stylesheet formatiert um (für die darin liegenden Elemente) ein gewünschtes Verhalten/Aussehen zu bekommen

– Browser fügen per default vor und nach einem einen 'Linebreak' ein.

• **Der HTML Code ist linear!**

• **Die Anordnung des Textes (der div-Blöcke) im HTML-Code bestimmt**

– wie im Screenreader vorgelesen wird

– Die ‚normale' TAB-Reihenfolge - Wenn man mit der Tastatur navigiert, dann kann man mit der ‚TAB' Taste von Link zu Link springen.

• Das <**table**> tag definiert eine HTML Tabelle.

– Hat immer ein oder mehrere <tr> <th> <td> Elemente.

– <tr> table row - Tabellenzeile

– <th> table head - Tabellenkopfzelle (vergleiche <thead>) th auch ‚vertikal' möglich

– <td> table data – Tabellenzelle

• Beachte:

– Zweidimensionale Inhalte sind schwer (nicht) zu linearisieren

– ‚summary‘ Attribut für textuelle Zusammenfassung der

• **Zusätzliche Tabellen-tags**

– **<thead> <tfooter> <tbody>** gruppieren Tabellenzellen im header, footer und body

– **<col>, <colgroup>** um Spalten (columns) besonders zu markieren und zu gruppieren

– Man spart sich Formate für jede Zelle (Muss vor <thead> kommen)

– <caption> für eine Tabellenüberschrift (-unterschrift)

```
<table>
<caption align="bottom">Tabellen UNTERschrift</caption>
</table>
```

– Das <**img**> tag definiert ein Bild (image):

– Hat 2 Pflichtattribute : "**src**" und "alt".

– Technisch gesehen werden Bilder nicht in eine Website eingefügt, sondern es wird zu einem Bild verlinkt (mit dem src Attribut).

– Das img Tag definiert den Platz an dem dieses Bild dargestellt werden soll.

– im alt Attribut wird ein alternativer (das Bild beschreibender) Text angegeben.

– Ist ein inline-block Element, erzeugt keinen Zeilenumbruch hat aber Höhe und Breite (des Bildes, sinnvoll anzugeben). Bilder sollte man nicht vom Browser skalieren lassen; width und height sollten die echten Dimensionen (in Pixel) des Bildes sein!

• Bilder die zugleich ‚Link‘ sind (also klickbar)

– "**alt**" Attribut (zumindest leer).

– evtl. zusätzlicher Text der ebenfalls klickbar ist.

• Das <**object**> tag definiert eingebundene Objekte innerhalb eines HTML Dokuments.

– Verwendet zum Einbinden von Multimedia Anwendungen (Audio, Video, Java Applets, ActiveX, PDF, Flash...)

– Mit dem <param> tag können Werte an das 'plugin' übergeben werden welches im <object> aufgerufen wird.

– Mindestens das "data" oder "type" Attribut muss angegeben werden.

– Für Barrierefreiheit muss erstens der <object> Code den Richtlinien entsprechen und zweitens das damit eingebundene Objekt zugänglich sein (z.b. der flash-player)

• **CSS** Cascading Style Sheets

– Styles definieren wie HTML Elemente dargestellt werden

– Typischer Weise in externen .css Files gespeichert (Zentrale Änderung der Darstellung möglich) <link href="files/styles.css" rel="stylesheet" />

– aber auch intern im HTML head <style type="text/css">...</style>

– aber auch inline in HTML Elementen <h1 style="background-color:red;">

• **'Cascading'**
– Mehrere Definitionen werden nacheinander aufgerufen und können sich auch (teilweise) überschreiben
• Reihenfolge:
– Browser default (kann von Browser zu Browser unterschiedlich sein)
– Externes Style Sheet (können auch mehrere sein, file.css)
– Interne Style
– Inline Style
Syntax:
• Eine Style-Regel besteht aus
  – Einem Selektor
  – Einer oder mehrerer Deklarationen (Eigenschaft-Wert Paare)
• Neben den HTML-Elementen können auch eigene Selektoren definiert werden: class und id wirken auf HTML Elemente mit diesen Attributen
• **# id – Zur Definition des Styles EINES einzigen Elementes**
• **. class – Zur Definition des Styles einer GRUPPE von Elementen**

• **Box Model**
– HTML Elemente werden als ‚Box' behandelt
• Margin - Außenabstand; Abstand vom Rahmen zu den umgebenden Elementen, hat keine Hintergrundfarbe, ist transparent
• Border - Rahmen; Kann Farbe und Dicke haben
• Padding - Innenabstand; Abstand vom Inhalt zum Rahmen, hat Hintergrundfarbe,
• Content – Inhalt; Hier ist Text oder Bild
• Nicht nur das Aussehen sondern auch die Anordnung (auf dem Bildschirm) von HTML Elementen sind über Styles definierbar.
– Sichtbarkeit: display:none visibility:hidden
– Positionierung
• Static - Default
• Fixed - relativ zum Browser-Fenster positioniert (bewegt sich beim scrollen nicht)
• Relative - relativ zur normalen Position des Elementes
• Absolute - wird an einer fixen Position im Browserfenster verankert (tatsächlich etwas komplizierter); kann andere Elemente überlappen: position:fixed;

• Mit Hilfe der Angabe eines **accesskey** Attributes (in dem ein Character angegeben wird) kann man über Tastatureingabe (Tastenkombination) innerhalb des HTML Dokumentes

an definierte Stellen springen.
• Welche Tasten verwendet werden können bestimmt der Programmierer
• Accesskeys sind in verschiedenen Browsern mit verschiedenen Tastenkombinationen realisiert z.B. • FF: Shift-Alt Character, Opera: Shift-ESC Character, Safari: Alt Character, Chrome: Alt Character
• Liste von Accesskeys ist (soll sein) zu Beginn eines HTML Dokumentes (um innerhalb des Dokumentes zu springen)
– Wird von Screenreadern als erstes gelesen
– Kann mit CSS verborgen werden, ist aber trotzdem als Information vorhanden
– Ohne CSS sieht man (wie Screenreader) die Accesskeys gleich zu Beginn des Dokumentes
Interne Links Zur Navigation
• Über den internen Link Sprung zum Element mit der entsprechenden id (in HTML5 kann jedes Element angesprungen werden)
• 'Unsichtbar' machen der Accesskey-Liste: Im CSS (eine von vielen Möglichkeiten) overflow: hidden}
• **Accesskeys für alle**
– Accesskeys nicht nur für blinde User -> beim „Durchtabben“ sichtbar machen (mit onFocus).
• Es gibt keinen Standard dafür, welche Tasten für accesskeys verwendet werden (sollen).
• Es kann zu Problemen kommen, wenn Tastenkombinationen bereits vom Betriebssystem oder Browser (oder auch von assistierender Technologie) belegt sind.

• Das **tabindex** Attribut definiert die TAB-Reihenfolge von Elementen (wenn die Tabulator-Taste zur Navigation verwendet wird)
• Default: Alle Links (und einige andere <button>, <input>...) Elemente werden in der Reihenfolge ihres Auftretens im HTML-Code mit der 'Tab'-Taste durchlaufen. (in HTML5 können alle Elemente angesprungen werden)
• Mit dem tabindex kann eine beliebige Reihenfolge definiert werden Elemente ohne tabindex kommen nach denen mit tabindex (sehr vorsichtig verwenden)
W3C
• tabindex = 0 – Das Element wird in die normale TAB-Order aufgenommen.
• tabindex= -1 – Das Element wird programmatisch fokusierbar (mit Script) aber wird nicht in die TAB-Order aufgenomme

HTML5

General

– Wenn Dokumente mit MIME-Type **"text/html"** geliefert werden, gelten sie als HTML-Dokumente und werden mit dem **HTML-Parser** verarbeitet. (als HTML5 bezeichnet).
– Wenn Dokumente mit MIME-Type **"application/xhtml+xml"** geliefert werden, dann gelten sie als XML-Dokumente und werden mit dem **XML-Parser** verarbeitet. (als XHTML5 bezeichnet).
– Beide besitzen ein gemeinsames DOM
• Doctype nur mehr **<!DOCTYPE html>**
• Wurzelelement leer (oder sehr kurz) **<html lang="en">**
• Charset 'kurz' **<meta charset="UTF-8">**
• **Syntax Problematik**: In XHTML gab es Regeln, welche eingehalten werden mussten.
  – Wohlgeformt
  – Elemente müssen End-Tags haben (auch leere Elemente!)
  – Tags müssen klein geschrieben werden
  – Keine leeren Attribute

• Aber in HTML5
– Großbuchstaben in Tag-Namen
– Anführungszeichen in Attributwerten optional
– Schließen von (leeren) Elements ist optional
Daher 'unsauberes' Programmieren sehr leicht möglich! Die Qualität des Codes wird unweigerlich leiden.

Validierung HTML5 vs. XHTML (Transitional)

http://html5.validator.nu/

http://validator.w3.org/

Neue semantische Elemente in HTML5

– Elemente welche die Dokumentstruktur verbessern sollen.
– Ersetzten (teilweise) Konstrukte
– Ob das Tag aber inhaltlich korrekt verwendet wird ist, wie bei den div-Attributen, Sache der Autorin des Webdokumentes

<section> – Allgemeiner Abschnitt/Bereich innerhalb eines Dokuments. Eine thematische Gruppierung von Inhalten (üblicherweise mit einer Überschrift)

<article> – Abgeschlossene Einheit eines Dokumentes . Z.B. Ein Foreneintrag, ein Zeitungsartikel, ein Benutzerkommentar.

<aside> – Information die nur losen Bezug zum Seiteninhalt hat. Inhalte die getrennt vom eigentlichen Seiteninhalt dargestellt werden können; Werbung/Leiste

<nav> – Repräsentiert einen Abschnitt des Dokumentes, der für Navigation bestimmt ist. (Nicht alle link-Gruppen müssen mit diesem Element zusammengefasst werden)

<header> – Repräsentiert eine Gruppe von Einführungselementen; (mit h1- Überschrift oder hgroup Elementen); Logo; Header für Dokument oder Sektion. (nicht zu verwechseln mit dem <head> tag)

<footer> – Footer für Dokument oder Sektion; Information über Autor, Copyright etc.

<mark> – Textverlauf der markiert oder hervorgehoben ist (highlight)

<time> <progress> <meter>... Tags dürfen selbstverständlich Attribute haben <head id="kopf"> ('redundant' aber natürlich erlaubt)

<figure> <figcaption> – Bilder, Diagramme, Fotos (aber auch Videos) und dazugehörige Unterschrift

<acronym> Abkürzung (jetzt abbr)

<big> großer Text

<tt> Teletext

<frame>, <frameset>, <noframe> „Rahmen“

<bb> Browser Button

<marquee> scrollendes Banner

<hgroup> Header Group; kam in HTML 5 und ist wieder verschwunden

<command> Kam in HTML 5 und ist wieder verschwunden

Video

• Videos mit 'Plugin' anzeigen – RealPlayer, QuickTime, Flash
• Probleme: Verfügbarkeit des Plugins (nicht jeder hat es installiert, nicht jeder erlaubt es), Accessibility -> Plugin muss accessible sein
• HTML5 definiert neues Element **<video>**
– Standardverfahren um Videos in Webseiten einzubinden
– Video wir durch den Browser und nicht durch ein externes Tool (Plugin) angezeigt
• **Aber**– Unterschiedliche Browser/Versionen unterstützen unterschiedliche Video-Formate (eventuell auch nicht vollständig)
Verschiedene **Video-Container-Formate**: In einem Container sind Videospur, Audiospur(en), Metadaten ... enthalten:
– MPEG4 - .mp4 (m4v); Basiert auf QuickTime Container
– Flash Video - .flv; 'Adobe'-Container
– Ogg - .ogv; Offener Standard, open source
– WebM – .webm; rel. neu, Ähnlichkeit mit MKV. Ausschließlich für VP8 Video und Vorbis Audio
• Verschiedene **Codecs**
– 'Codec' ist ein Algorithmus gemäß dem ein Videostream (bzw. der Audiostream) kodiert ist.
– Verschiedene Stream-Formate innerhalb der Container;
• **Video**:
– H.264 ; MPEG-4 AVC (Advanced Video Coding) patentbelastet
– Theora; Kostenloser Codec. Am häufigsten in Ogg Containern
– VP8; Kostenfrei von Google
• **Audio Codecs**
– MPEG-1 Layer 3 – mp3; patentbelastet
– AAC ; Advanced Audio Codec, patentbelastet
– Vorbis; open source
• **Kombinationen Container-Codec**
– Theora+Vobis+Ogg
– H.264+AAC+MP4
– VP8+Vorbis+WebM
• Auf jeden Fall **Prüfen welche Container/Codecs in welchen Browsern funktionieren** (MP4 dzt. in so gut wie allen (aktuellen) Browsern möglich)

Wenn man ein Video auf allen Plattformen verfügbar machen will, dann muss man es eventuell in unterschiedlichen Formaten kodieren und anbieten (Lizenzgebühren beachten)

• **Anwendung in HTML5**

– **Minimal** <video src="myvideo.webm"></video>

– **Sinnvoll** <video src="myvideo.webm" width="320" height ="240" other-attributes ></video>

• Verschiedene Formate innerhalb des <video> tags anbieten. Browser geht die Liste der Quellen durch und spielt das erste Video welches er darstellen kann.

• Um zu verhindern, dass Browser Videos laden die sie nicht darstellen können, sollen entsprechende Attributwerte verwendet werden

• Achtung: Videodateien müssen mit dem korrekten MIME-Type ausgeliefert werden!

• Für ältere Browser 'eine Alternative angeben (siehe „Flash“)?

• HTML5 legt fest, dass Kind-Elemente des video-Tags ignoriert werden sollen (außer <source>).

• Daher kann man ein Objekt einbinden, welches von neuen Browsern ignoriert wird, während alte Browser das <video> ignorieren.

¤ Flash für Videos sollte aus Technologiegründen und Zugänglichkeitsüberlegungen vermieden werden!

Accessibility

• Mit dem nativen **'controls'** Attribut sollte 'accessible' Zugriff auf die Steuerelemente möglich sein.

– Mit **'Tab'** zu jedem Element (play, progress bar, volume)

– **Leertaste** für play/pause toggle

– links – rechts Pfeile für **Navigation** (im Progressbar)

– hinauf – hinunter Pfeile für **Lautstärke**

• Für einheitliche Bedienbarkeit und einheitliches Aussehen CSS, JavaScript (?)

• Kontextmenü, Untertitel, Audiodescription, noch sehr unterschiedlich gehandhabt bzw. nicht implementiert (oder mit JavaScript)

Formulare XHTML

• Das <**form**> tag:

– Definiert Formulare um Benutzereingaben möglich zu machen (und diese an einen Server zu übermitteln)

– Innerhalb das <form> tags sind folgende tags (mehrfach) möglich

<input>, <textarea>, <button>, <select>, <option>, <optgroup>, <fieldset>, <label>

• Identifikation eines Elements in einem Formular: **HTML** das **'name'** Attribut, **XHTML** stattdessen **'id', HTML5** ist wieder **'name'**
• Im <form> mit dem **'action'** Attribut angeben was mit den Benutzerdaten passiert: <form action="http://www.meineseite.at/doit.php" method="post">
• Eingabefeld <Formulare XHTML> – Verschiedene 'Typen' von Eingaben mit dem type Attribut. • <input type=...>
– **text** : Eine Inputzeile für Texteingabe.
– **password** : Eine Zeile für 'Passworteingabe'. Achtung die Zeichen werden nur maskiert!
– **checkbox**
– **radio** : Radiobuttons
– **submit** : Ein 'Submit' Button. Beim Aktivieren des Buttons (Klick, Enter) werden die Formulardaten an den Server gesendet.
• **Sehr viele (neue) Attribute in HTML5**
• Für <**form**>: – autocomplete, novalidate
• Für <**input**>: – autocomplete, autofocus, form, formaction, formenctype, formmethod, formnovalidate, formtarget, height, width, list, min, max, multiple, pattern (regexp), placeholder, required, step
• **autofocus**: Wenn das Formular geladen wird bekommt das Element automatisch Fokus
• **pattern**: 'Muster' gegen das die Eingabe geprüft wird
• **required**: Eingabe notwendig - Bei Verwendung von 'required' verlagert man die Verantwortung auf den Browserhersteller
• **novalidate**: Validierung ausgeschaltet
• **Kein Javscript oder serverseitige Überprüfung notwendig**
• Jedes Element braucht ein **'label'**! (quickcheck: Klick auf den Text (das Label), und das dazugehörige Element wird fokussiert) <label for="vorname">Vorname *</label>
• Radiobuttons und Checkboxes brauchen **<fieldset> und <legend>**

Accessibility

• Eingabeformat angeben: <label for="date">Datum TT/MM/YYYY *</label>
• Verschiedene Browser (und verschiedene Versionen des selben Browsers) haben HTML5 Funktionalität unterschiedlich weit implementiert
• Wer kann was? (z.B. https://modernizr.com)
• Javascript Bibliothek
– In die Website einbinden
– Testet HTML5-Fähigkeit des Browsers

• HTML5test - how well does your browser support HTML

PDF als zugängliches Dokument

General

Ursprünglich Postscript für Laserdrucker, Quasistandard bzw. allg. Druckvorstufe.
Weitergabe von Postscript-Dokumenten in wissenschaftlichen Kreisen.
Nachteile von Postscript:

– Sehr große Dateien
– Ausschließlich für Druck gedacht und entwickelt
– Zugriff auf und Änderung der Dateien sehr schwer

PDF Vorteile

- Dokumentenformat und nicht mehr Grafikprogrammiersprache
- Betriebssystem- und hardwareunabhängig
- Auch für direktes Lesen gedacht
- Interaktive Formulare / skriptfähig
- Einfach zu bedienende Gratisviewer für alle
- Betriebssysteme und Plattformen ➔ weit verbreitet

PDF Probleme

- Zunächst war es ein unabhängiges Format für Dokumente, vorwiegend zum Drucken
    - -> Es war nur interessant wie es aussieht!
    - -> PDF nur so zugänglich wie ein gedrucktes Papier!
- Da PDF das Papier ersetzen soll/sollte wurden Verbesserungen eingefordert, die die Zugänglichkeit ermöglichen sollen

Forderungen an zugängliches PDF

- Text direkt zugreifbar
- Kennzeichnung von Texteigenschaften wie Sprache
- Navigationsmöglichkeiten im Dokument
- Lesereihenfolge von Text im Dokument
- Strukturierung von Text (Überschriften, Aufzählungen, ...)
- Alternative Beschreibungen für Grafiken, Bilder, ...
- Orientierungshilfen / Strukturerläuterungen für Tabellen
- Beschriftung von Formularfeldern
- Beschriftung von interaktiven Elementen (z.B. Buttons)

PDF Normung

- Zunächst proprietär, alle Programme kostenpflichtig
- Danach Offenlegung der Strukturen aber weiterhin alle Rechte vorbehalten
- ISO 15930 PDF/X zur Übermittlung von Druckvorlagen
- ISO 19005 PDF/A für Langzeitarchivierung
- ISO 24517-1 PDF/E, PDF/H spezielle Fassungen für technische Dokumente und Gesundheitswesen
- ISO 32000-1:2008 PDF 1.7 -> offener Standard
- **ISO 14289-1:2014 Standard einer normierten Zugänglichkeitsbeschreibung -> PDF/UA**

2009 ISO übernimmt PDF/UA Projekt
2012 erstmals als ISO Norm verabschiedet

Enthält
– Einleitung, Begriffsbestimmungen, ...
– Forderungen an das Dokument (und damit den Autor)
(Forderungen an PDF Reader und AT)

Voraussetzungen
– Getagtes PDF nach ISO Norm

Forderungen an das Dokument
– Trennung von Text und Artefakten (Bedeutungstragend vs. Dekorativ)
– Bedeutungstragende Inhalte müssen mit semantisch geeigneten Tags versehen werden
– Artefakte werden als solche gekennzeichnet
– Angabe der logischen Lesereihenfolge (Strukturbaum)
– Es dürfen nur Tags verwendet werden die in PDF1.7 definiert sind (Standard Tags); Anderen Tags muss eine entsprechende Rolle zugewiesen werden
– Informationen dürfen nicht durch rein visuelle Mittel (Kontrast, Farbe, Anordnung) vermittelt werden.
– Flackernde, blitzende, blinkende Inhalte dürfen nicht vorkommen
– Ein Dokumententitel muss vorhanden sein
– Angabe der Sprache, Kennzeichnung von Sprachwechsel
– Grafiken als solche kennzeichnen, aber nur wenn sie sinnvollen Inhalt bieten. Sonst ist es ein Artefakt

– Bildbeschreibungen (Captions) als solche kennzeichnen
– Überschriften verwenden, Reihenfolge beachten und mit H1 beginnen
– Tabellen als solche kennzeichnen und Zeilen/Spaltenüberschriften kennzeichnen
– Listen kennzeichnen und wirklich nur für entsprechende Texte benutzen
– Formeln kennzeichnen
– Links kennzeichnen
– Signaturen sollen zugänglich sein, darin enthaltene Grafiken brauchen einen Alt-Text
– Alle embedded oder attached files sind per Definition auch zugänglich zu machen

Matterhorn Protokoll

- Wurde vom PDF/UA Competence Center der PDF Association ausgearbeitet. (https://www.pdfa.org/)
- Ist eine Übersetzung des ISO-Standards in 31 Prüfpunkte mit 136 Fehlerbedingungen.
- 89 dieser Fehlerbedingungen lassen sich maschinell prüfen
- 45 kann nur ein Mensch beurteilen.
- Bei 2 Fehlerbedingungen gibt es keinen spezifischen Test. (Signaturen, Navigation)
- Tests für einzelne Objekte wie Tags, Tabellen oder Links, für einzelne Seiten und das ganze Dokument.

Logische Dokumentenstruktur & Semantik
Vollständig: Jedes Element hat einen Tag oder ist als Artefakt gekennzeichnet
Linear: alle Inhaltselemente sind in einer eindeutigen Reihenfolge
Adäquat: jedes Inhaltselement hat das zu ihm passenden Tag
Kongruent: was zusammen gehört, muss zusammen bleiben: Umgang mit Trennstrichen, Zeilen oder Seitenumbrüchen
Eindeutig: nur PDF-Standard Tags verwenden oder benutzerdefinierten Tags einen solche zuordnen (Rollenzuordnung)

Tags in PDF

- Struktur und Zusatzinformationen ähnlich (bis identisch) wie bei HTML
- Vorhandensein von Tags ist quasi ein Muss für die Zugänglichkeit aber i.Allg. nicht ausreichend
- Die Struktur des PDF Dokuments muss ebenfalls stimmen, um überhaupt sinnvoll getagt zu werden

Erstellen von PDFs

Erstellen von getagten PDFs aus entsprechender Source -> Im Allgemeinen recht einfach und schnell. Erstellen von getagten PDFs aus ungetagtem PDF (d.h. aus 'schlechter' Quelle) -> Im Allgemeinen EXTREM mühsam

Source - PDF Quellen

- Alle Informationen für das zugängliche PDF müssen schon in der Source sein!
- Strukturierte Überschriften, Inhaltsangabe
- Formatvorlagen verwenden
- Alternativtexte für Grafiken, Bilder
- Keine Leerzeichen und Leerzeilen für Positionierungen verwenden
- Hyperlinks (korrekt) verwenden (Verweise, Index,...)
- Am besten macht man zugängliche Quellen (zugängliche Word Dokumente, Präsentationen, ...) um draus zugängliche PDFs zu machen.
- In zugänglichen Dokumenten sind sicher alle (sehr viele) Informationen enthalten. Diese Dokumente können auch gut "upgedated" werden (und bleiben dann auch zugängliche Quellen).
- Die Zugänglichkeit von Microsoft Dokumenten kann man mit dem integrierten Checker überprüfen

PDFs erstellen

- **Microsoft Word**
  – Ab Version 2007 ist die PDF Exportfunktion integriert
  – Für alle Versionen Adobe Plugin (käuflich)
  – PDF Druckertreiber sind NICHT geeignet (da die Dokumentenstruktur verloren geht)
- **Open Office**
  – Ab Version 2 können tagged PDFs erzeug
- **Adobe Framemaker**
- **Adobe Illustrator**
- **Adobe InDesign**
- **Adobe LiveCycle Designer (Formulare)**
- **Immer noch erzeugen viele Programme unzugängliche PDFs (der Autor ist machtlos)**

Microsoft Word

- Alt Texte bei folgenden 'Objekten' verwenden: Bilder, Clipart, Charts, Formen, SmartArt, Gruppen, Embedded Objekte, Video und Audio Dateien
- Tabellen Header definieren – In den Tabellentools Entwurf -> Überschrift anhaken / Layout -> Überschrift wiederholen
- Überschriften Vorlagen in der richtigen Schachtelung verwenden
- Eindeutigen Text für Links verwenden
- Möglichst simples Tabellenlayout verwenden
- Verwende keine leeren Paragraphen oder mehrere Leerzeichen
- Vermeide 'floating' Objekte, verwende 'mit Text in Zeile'.
- Vermeide 'Wasserzeichen'
- **Konvertierung**
    - Kopf- und Fußzeilen werden als Hintergrund gespeichert -> nicht in der Dokumentenstruktur enthalten
    - Tabellen müssen z.T. im PDF nachbehandelt werden (TH werden evtl. nicht gesetzt)
    - Spracheinstellungen müssen im PDF kontrolliert und möglicherweise nachbehandelt werden
    - Mehrspaltige Seiten mit den entsprechenden Einstellungen und nicht mit Tabellen
    - Wo immer es geht Standardformate verwenden
    - Leitfaden BMASK

Microsoft PowerPoint

- Alles beachten was in Word zu beachten ist
- Lesereihenfolge überprüfen – Start / Zeichnung / Anordnen → Auswahlbereich
- **Konvertierung**
    - Alle Inhalte aus dem Folienmaster werden zu Hintergrund und sind nicht in der Struktur enthalten
    - Tabellen müssen mit Headern nachgebessert werden

Kontrolle von PDF Dokumenten

- Check mit Adobe Acrobat
- Keine Probleme beim Adobe Acrobat Check bedeutet NICHT AUTOMATISCH auch zugänglich -> Händische Checks notwendig

- Externe Tools (PDF/UA) – PAC 3 (PDF Accessibility Checker v3)
- **Händische Checks**
    - Die Dokumenteneigenschaften zeigen, ob das Dokument getagged ist
    - Mit dem erweiterten Werkzeug TouchupLeserichtungswerkzeug die Dokumentenstruktur überprüfen
    - Bei Bedarf sich Seiten vorlesen lassen

Reparatur von PDF

Verschiedene Werkzeuge in Adobe Acrobat (z.T. Pro Edition erforderlich)

– Touchup-Leserichtungswerkzeug
– Reihenfolge-Fenster (Inhalt, Reihenfolge, Tags)
– Tabellen-Editor (Zellen-Tags, Summary, ...)
– Die meisten Werkzeuge sind per Default ausgeblendet

Versuch Text durch OCR erkennen zu lassen ODER automatisch taggen zu lassen.

Kommerzielle(!) Tools: axesPDF for Word, axesPDF QuickFix, Foxit (gratis Reader, kostenpflichtiger Editor)

PDF und Latex

LaTeX im universitären Bereich beinahe/manchmal 'ein Muss' um Dokumente zu verfassen (z.B. Masterarbeiten). Es gibt (derzeit) keine Möglichkeit aus LaTeX ein PDF/UA Dokument zu erzeugen. Einige Lösungsansätze: LaTeX Pakete die zur accessibility beitragen (accessibilityMeta, babel, hyperref)

Gut überlegen, ob es überhaupt Sinn macht PDF auf Web-Seiten zu verwenden.

Reicht nicht auch "normales HTML"?

WCAG

WCAG1

General

Richtlinien wie Webinhalte behinderten Menschen zugänglich gemacht werden können (1999).
Richten sich an Entwickler von Web-Inhalten und an Entwickler von Tools zur Seitenerstellung.
-> Förderung der Zugänglichkeit, von WAI veröffentlicht
Zugang für alle Benutzer verbessert:
– unterschiedliche ‚user-agents' (Browser)
– Umgebungsbedingungen (Bedienung...)

Struktur

• **14 Richtlinien** (Hauptvorgaben) – Unterteilt in **65 Checkpunkte**
• **3 Prioritätsstufen**
Priorität 1 – Webinhalt **MUSS** diesen Punkt erfüllen, Nichterfüllung macht es **bestimmten Benutzergruppen unmöglich** Webdokumente zu verwenden
• Priorität 2 – Webinhalt **SOLLTE** diesen Punkt erfüllen, Nichterfüllung macht es **einzelnen Gruppen schwer** die Web-Dokumente zu verwenden
Priorität 3 – Webinhalt **KANN** diesen Punkt erfüllen, Nichterfüllung macht es **einzelnen Gruppen etwas schwierig** die Web-Dokumente zu verwenden
3 Konformitätsstufen= **A (Checkpunkte der Priorität 1), AA (1 & 2), AAA (1 & 2 & 3)**

Richtlinien

1) **Äquivalente Alternativen für Audio- und visuellen Inhalt bereit.**

- Audio-Inhalt und Bilder textuell beschreiben 1
- Zusätzlicher Inhalt 1
- Quickcheck: Bilder abschalten; Seite vorlesen (lassen) 1
- redundante Textlinks, Untertitel synch 1

2) **Verlassen Sie sich nicht auf Farbe allein.**

- Farbfehlsichtigkeit und Benutzer ändern die gewusst 1
- Information auch ohne Farbe verfügbar 1
- Quickcheck: Farben bewusst auf S/W stellen und die Seite dann benutzen
- Vordergrund- und Hintergrundfarbe ausreichend kontrastieren 2-3

3) **Verwenden Sie Markup und Stylesheets auf korrekte Weise.**

- Markup anstelle von Bildern 2
- Dokumente, die gegen veröffentlichte formale Grammatiken validieren 2
- Richtlinien Stylesheets, um Layout und Präsentation (spacing+firstletter) 2
- relative anstelle von absoluten Einheiten 2
- Überschriften-Elemente verwenden 2
- korrekten Markup für Listen und Listenelemente 2
- Markup für Zitate, nicht um visuelle Effekte wie Einrückung zu erzielen

4) **Verdeutlichen Sie die Verwendung natürlicher Sprache.**

- Änderungen kenntlich machen aufklarer Weise – Verwende "lang" Attribut 1
- Ausschreibung jeder Abkürzung und Akronyms an der Stelle des ersten Auftretens 3
- vorherrschende natürliche Sprache des Dokuments kenntlich machen 3

5**) Erstellen Sie Tabellen, die geschmeidig transformieren.**

- Tabellen nicht für Seitenlayout
- Kennzeichne Zeilen- und Spaltenüberschriften

6) **Seiten, die neue Technologien verwenden, geschmeidig transformieren.**

- neue Technologien aber Inhalte bleiben zugänglich bleiben wenn diese abgeschaltet werden oder ältere Browse verwendet werden
- Wenn Javascript abgeschaltet ist, muss die Seite immer noch funktionieren, wenn das nicht geht muss ein passendes <noscript> Element vorhanden sein **(NOT WCAG2)**

7) **Kontrolle des Benutzers über zeitgesteuerte Änderungen des Inhalts.**

- Bewegte, scrollende oder sich automatisch ändernde Objekte oder Seiten müssen angehalten oder gestoppt werden können. Menschen mit visuellen Einschränkungen können bewegten Text evtl. nicht schnell genug lesen,

Menschen mit physischen Behinderungen können evtl. nicht schnell genug reagieren 1
- Vermeiden Sie Bildschirmflackern - Epileptische Anfälle 2
- vermeiden Sie es, Inhalt blinken zu lassen 2
- Vermeiden Sie Bewegung in Seiten, bis Benutzeragenten das Einfrieren von Bewegung ermöglichen 2
- keine Seiten mit automatischer periodischer Aktualisierung (http-equiv="refresh") 2
- verwenden Sie keinen Markup, um eine Weiterleitung zu erzielen -> Server 2

8) **Direkte Zugänglichkeit eingebetteter Benutzerschnittstellen.**

- die Schnittstelle muss zugänglich sein oder eine alternative zugängliche Lösung 1-2

9) **Geräteunabhängiges Design. (**siehe Hilfsmittel)
10)**Verwenden Sie Interim-Lösungen.**

- vorläufig, damit assistive Technologien und ältere Browser korrekt funktionieren

11)**Verwenden Sie W3C-Technologien und -Richtlinien.**

- oder alternative Version des Inhalts bereit, die zugänglich ist

12)**Informationen zum Kontext und zur Orientierung bereitstellen.**

- Informationen um Benutzern das Verständnis komplexer Seiten oder Elemente zu erleichtern.
- Komplexe Beziehungen zwischen Teilen einer Seite sind möglicherweise schwer zu interpretieren

13)**Klare Navigationsmechanismen bereitstellen.**

- eine Person auf der Seite findet was sie sucht
- Ziel des Links 2 - Linktext soll aussagekräftig sein, Linktext soll knapp sein, zusätzlich ‘title‘ Attribut. – Vermeide: ‘Hier klicken‘
- Metadaten bereitstellen, um semantische Information zu Seiten und Sites 2
- Informationen zum allgemeinen Layout einer Site bereit (z. B. über eine Sitemap oder ein Inhaltsverzeichnis) 2
- Verwenden Sie Navigationsmechanismen in konsistenter Weise 2

- Informationen zum allgemeinen Layout einer Site bereit (z. B. über eine Sitemap oder ein Inhaltsverzeichnis) 2
- Verwenden Sie Navigationsmechanismen in konsistenter Weise 2
- Verwenden Sie Navigationsmechanismen in konsistenter Weise 3
- Gruppieren Sie verwandte Links, identifizieren Sie die Gruppe (für Benutzeragenten), und ermöglichen Sie das Überspringen der Gruppe, bis Benutzeragenten dies gestatten. 3 -> accesskey
- verschiedene Arten der Suche 3
- unterscheidungskräftige Information an den Anfang von Absätzen, Listen 3
- Informationen über Zusammenstellungen von Dokumenten 3
- Überspringen von mehrzeiligen ASCII-Zeichnungen 3

14)**Dokumente sind klar und einfach gehalten.**

- Konsistentes Seitenlayout, deutliche Grafiken, leicht verständliche Sprache 1
- Inhalt einer Site die klarste und einfachste Sprache 1
- Ergänzen Sie Text mit grafischen oder Audio-Präsentationen 3
- Präsentationsstil, der über Seiten hinweg konsistent ist (Externes CSS, Durchdachte Navigation)

Validierung

Verwenden von automatisches Zugänglichkeits-Tool und Browser-Validierungs-Tool für Syntax und Stylesheets. Opt einen Test-Browser oder Emulator verwenden.
Verwenden Sie mehrere Grafik-Browser, alte und neue Browsers, sprachgenerierenden Browser, einen Screenreader, Vergrößerungs-Software, ein kleines Display.
Rechtschreib- und Grammatikprüfung -> verbessert das Verständnis
Klarheit und Einfachheit -> Korrekturleser verbessern Verwendbarkeit
Fordern Sie behinderte Menschen auf, Dokumente zu überprüfen.

WCAG2

General

2008 - Lange Diskussion um die technische Sinnhaftigkeit – Große Differenzen zwischen verschiedenen Usergruppen und Technologievertretern
Darüber hinaus wird das Befolgen dieser Richtlinien Ihre Webinhalte in vielen Fällen für Nutzer im Allgemeinen benutzbarer machen.

= **ISO/IEC 40500:212**

Weg von der technologischen (HTML) Sicht von WCAG 1 => technologie-neutral

WCAG 2.1 (2018)

WCAG 2.1 ist 'rückwärtskompatibel sein' mit WCAG 2.0
Technik (**Webinhalt**) = **Mechanismus zum Kodieren von Anweisungen, die von Benutzeragenten gerendert, abgespielt oder ausgeführt werden sollen**.

4 Prinzipien

Zu jedem Prinzip gibt es Richtlinien. Jede Richtlinie wird mit meßbaren Erfolgskriterien überprüfbar gemacht. Jedes Prinzip und jede Richtlinie wird durch ausreichende und empfohlene Techniken untermauert. (um die Barrierefreiheit zu erreichen)

KONFORMITÄT

Zu einem Standard konform sein heißt seine Erfolgskriterien erfüllen = wichtige Zugänglichkeitskriterien für Menschen mit Behinderungen. Müssen überprüfbar sein, von Menschen die Erfolgskriterien verstehen, wie Behinderte mit Webtechnologien umgehen oder wissen, wie diese die benutzen.

A, AA, AAA - Konformität kann nur für eine komplette Seite erreicht werden

Alle Seiten müssen den Minimumstandard erfüllen. Es wird nicht empfohlen, Konformität auf Stufe AAA als allgemeine Richtlinie für komplette Websites zu fordern Seiten, die nicht konform sein können (z.B. Testseiten) müssen aus dem Bereich der konformen Seiten ausgenommen werden

Autoren können WCAG 2.0-konform sein, ohne eine Erklärung abzugeben.
Konformitätslogo muss von den erforderlichen Komponenten einer **Konformitätserklärung** begleitet sein.

^**Informationen**: Datum der Erklärung, Titel der Richtlinien, Version und URI der Richtlinien, Erfüllte Konformitätsstufe: (Stufe A, AA oder AAA), eine präzise Beschreibung der Webseiten, wie zum Beispiel eine Liste von URIs, für welche die Erklärung abgegeben wird einschließlich der Information, ob Subdomains in der Erklärung beinhaltet sind, eine Liste mit Webinhalts-Techniken, auf die man sich verläßt.

Optionale Komponenten einer Konformitätserklärung. Eine Liste mit erfüllten Erfolgskriterien, die über die erklärte Konformitätsstufe hinausgehen. Diese Information sollte auf eine Art und Weise zur Verfügung gestellt werden, die Benutzer nutzen können, vorzugsweise maschinen-lesbare Metadaten. Eine Liste mit den speziellen Techniken.

Eine Liste mit Benutzeragenten einschließlich assistierender Techniken, die benutzt wurden um den Inhalt zu testen. Informationen zu allen zusätzlich unternommenen Schritten, die über das Erfolgskriterium hinausgehen, um die Barrierefreiheit zu verbessern. Eine maschinen-lesbare Metadaten-Version der Liste mit speziellen Techniken, auf die man sich verlässt. Eine maschinen-lesbare Metadaten-Version der Konformitätserklärung.

Prinzipien

- **Wahrnehmbar = Informationen und Bestandteile der Benutzerschnittstelle müssen den Benutzern so präsentiert werden, dass diese sie wahrnehmen können.**
    - **Textalternativen**
        - Großschrift, Braille, Symbole oder einfachere Sprache für alle Nicht-Text-Inhalte **A**
        - Eingabeelemente: Der Name gibt Aufschluß **A**
        - Zeitbasierte Medien: Zumindest eine Kurzbeschreibung **A**
        - Sensorisch: Zumindest eine Erklärung was es sein sollte **A**
        - CAPTCHA Alternativen anbieten **AA**
        - Dekoration **AA**
    - **Zeitbasierte Medien**
        - Aufgezeichnete Medien, Live Medien, Gebärdensprache für Audio-Inhalte
    - **Anpassbar**
        - Inhalte, die auf verschiedene Arten dargestellt werden können **A**
        - Info und Beziehungen können durch Software bestimmt werden oder stehen in Textform zur Verfügung **A**
        - die korrekte Leseabfolge durch Software bestimmt **A**
        - stützen sich nicht nur auf sensorische Eigenschaften von Komponenten **A**

- **Unterscheidbar**
    - Inhalt zu sehen und zu hören einschließlich der Trennung von Vorder- und Hintergrund **A**
    - Audio-Steuerelement - einen Mechanismus, um die Wiedergabe zu pausieren oder zu beenden, oder es gibt einen Mechanismus, um die Lautstärke unabhängig von der allgemeinen Systemlautstärke zu regeln **A**
    - Die visuelle Darstellung von Text und Bildern von Text hat ein Kontrastverhältnis von mindestens 4,5:1 **A**
    - Text ohne assistierende Technik um bis zu 200 Prozent geändert **A**
    - Text statt Bilder eines Textes dazu benutzt wenn nicht Anpassbar und Unentbehrlich **AA**
    - Text und Bildern eines Textes hat ein Kontrastverhältnis von mindestens 7:1 WENN nicht Großer Text, Nebensächlich, Wortbildmarken, keine Kontrastanforderungen **AA**
    - Für reinen Audioinhalt gilt: Kein Hintergrund, Abschalten, Hintergrundgeräusche abgeschaltet oder 20 dB leiser als Sprachinhalt **AAA**
    - Vorder- und Hintergrundfarben können vom Benutzer ausgewählt **AAA**
    - Breite beträgt nicht mehr als 80 Zeichen oder Glyphen **AAA**
    - Text ist nicht im Blocksatz ausgerichtet **AAA**
    - Der Zeilenabstand (Durchschuss) ist mindestens 1,5-fach innerhalb von Paragraphen und der Paragraphenabstand ist mindestens 1,5-fach so groß wie der Zeilenabstand. **AAA**
    - Die Textgröße kann ohne assistierende Technik bis auf 200 Prozent skaliert werden und zwar so, dass der Leser nicht horizontal scrollen muss **AAA**
    - Bilder eines Textes werden nur rein dekorativ benutzt oder dann, wenn eine bestimmte Präsentation von Text unentbehrlich für die zu vermittelnden Informationen ist **AAA**

- **Bedienbar = Bestandteile der Benutzerschnittstelle und Navigation müssen bedienbar sein**

- **Per Tastatur zugänglich:** Alle Funktionalitäten des Inhalts sind durch eine Tastaturschnittstelle bedienbar. **A**
- Wenn der Tastaturfokus durch eine Tastaturschnittstelle auf einen Bestandteil der Seite bewegt werden kann, dann kann der Fokus von diesem Bestandteil weg bewegt werden, indem man nur die Tastaturschnittstelle benutzt **A**
- Alle Funktionalitäten des Inhalts sind durch eine Tastaturschnittstelle bedienbar **A**
- **Ausreichend Zeit**: Geben Sie den Benutzern ausreichend Zeit, Inhalte zu lesen und zu benutzen:
  Zeitliche Begrenzung: Abschalten, Anpassen, Ausweiten;
  Ausnahme: Echtzeit-Ausnahme, Unentbehrliche Ausnahme, 20 Stunden-Ausnahme **A**
- Pausieren, beenden, ausblenden: Für sich bewegende, blinkende, scrollende oder sich automatisch aktualisierende Informationen **A**
- Die Zeiteinteilung ist kein unentbehrlicher Teil eines vom Inhalt dargestellten Ereignisses **AAA**
- Unterbrechungen können vom Benutzer aufgeschoben oder unterdrückt werden **AAA**
- Erneute Authentifizierung ohne Datenverlust **AAA**
- **Anfälle:** Gestalten Sie Inhalte nicht auf Arten, von denen bekannt ist, dass sie zu Anfällen führen. Webseiten enthalten nichts, was öfter als dreimal in einem beliebigen, eine Sekunde dauernden Zeitraum blitzt. **AAA**
- Webseiten enthalten nichts, das öfter als dreimal in einem beliebigen, eine Sekunde dauernden Zeitraum blitzt **AAA**
- **Navigierbar:** Stellen Sie Mittel zur Verfügung, um Benutzer dabei zu unterstützen zu navigieren, Inhalte zu finden und zu bestimmen, wo sie sich befinden **A**
- Blöcke umgehen **A**
- Seite mit Titel versehen: Webseiten haben einen Titel, der Thema oder Zweck beschreibt. **A**
- **Fokus-Reihenfolge**: Wenn eine Webseite der Reihe nach navigiert werden kann und die Reihenfolge der Navigation die Bedeutung oder Bedienung beeinflusst, erhalten fokussierbare Komponenten den Fokus in einer Reihenfolge, der Bedeutung und Bedienbarkeit aufrecht erhält. **A**
- **Linkzweck**: Der Zweck jedes Links kann durch den Linktext allein oder durch den Linktext zusammen mit seinem durch Software bestimmten Link-Kontext bestimmt werden **AAA**
- Verschiedene Methoden: Es gibt mehr als eine Methode, um eine Webseite innerhalb eines Satzes von Webseiten zu finden, außer die Webseite ist das Ergebnis oder ein Schritt innerhalb eines Prozesses. **AA**
- Überschriften und Beschriftungen (Labels): Überschriften und Labels beschreiben ein Thema oder einen Zweck **AA**
- Überschriften und Beschriftungen (Labels): Überschriften und Labels beschreiben ein Thema oder einen Zweck **AA**
- Position: Es gibt Informationen zu der Position des Benutzers innerhalb eines Satzes von Webseiten. **AAA**

    - Linkzweck (reiner Link): Es gibt einen Mechanismus, um den Zweck jedes Links durch den Linktext allein zu erkennen, außer der Linkzweck wäre mehrdeutig für Benutzer im Allgemeinen. **AAA**
    - Abschnittsüberschriften: Abschnittsüberschriften werden genutzt, um den Inhalt zu gliedern. **AAA**

- **Verständlich = Informationen und Bedienung der Benutzerschnittstelle müssen verständlich sein**
    - **Lesbar**
        - **Sprache der Seite**: Die voreingestellte menschliche Sprache jeder Webseite kann durch Software bestimmt werden. **A**
        - **Sprache von Teilen**: Die menschliche Sprache jedes Abschnitts oder jedes Satzes im Inhalt kann durch Software bestimmt werden außer bei Eigennamen, technischen Fachbegriffen, Wörtern einer unklaren Sprache und Wörtern oder Wendungen, die Teil des Jargons des direkt umliegenden Textes geworden sind **AA**
        - **Ungewöhnliche Wörter:** Es gibt einen Mechanismus, um spezielle Definitionen von Wörtern oder Wendungen zu erkennen, die auf ungewöhnliche oder eingeschränkte Weise benutzt werden, Idiome und Jargon eingeschlossen **AA**
        - **Abkürzungen**: Es gibt einen Mechanismus, um die ausgeschriebene Form oder Bedeutung von Abkürzungen zu erkennen **AAA**
        - **Leseniveau**: Wenn der Text nach der Entfernung von Eigennamen und Titeln Lesefähigkeiten voraussetzt, die über das Niveau der niedrigen, sekundären Schulbildung hinausgehen, dann gibt es ergänzenden Inhalt oder eine Version, die keine über die niedrige, sekundäre Schulbildung hinausgehenden Lesefähigkeiten geht **AAA**
        - **Aussprache**: Es gibt einen Mechanismus, um die bestimmte Aussprache von Wörtern zu erkennen, wenn die Bedeutung der Wörter - im Zusammenhang - mehrdeutig ist, wenn man die Aussprache nicht kennt **AAA**
        - **Vorhersehbar:** Webseiten vorhersehbar aussehen und funktionieren
            - Bei Fokus: Wenn irgendein Bestandteil den Fokus erhält, dann löst dies nicht eine Änderung des Kontextes aus. **A**
            - Bei Eingabe: Die Änderung der Einstellung irgendeines Bestandteils der Benutzerschnittstelle führ nicht automatisch zur Änderung des Kontextes **A**
        - **Konsistente Navigation**:Navigationsmechanismen, die auf mehreren Webseiten innerhalb eines Satzes von Webseiten wiederholt werden, treten jedes Mal, wenn sie wiederholt werden, in der gleichen relativen Reihenfolge auf, außer eine Änderung wird durch den Benutzer ausgelöst. **AA**
        - **Konsistente Erkennung**: Bestandteile mit der gleichen Funktionalität innerhalb eines Satzes von Webseiten werden konsistent erkannt. **(Stufe AA)**

    - **Änderung auf Anfrage**: Änderungen des Kontextes werden nur durch Benutzeranfrage ausgelöst oder es gibt einen Mechanismus, um solche Änderungen abzuschalten. **AAA**
    - **Hilfestellung** bei der **Eingabe** Helfen Sie den Benutzern dabei, Fehler zu vermeiden und zu korrigieren. **A**
    - **Fehlererkennung**: Wenn ein Eingabefehler automatisch erkannt wird, dann wird das fehlerhafte Element identifiziert und der Fehler wird dem Benutzer in Textform beschrieben. **A**
    - **Beschriftungen (Labels) oder Anweisungen**: Wenn der Inhalt eine Eingabe durch den Benutzer verlangt werden Beschriftungen (Labels) oder Anweisungen bereitgestellt. **A**
    - **Fehlerempfehlung**: Wenn ein Eingabefehler automatisch erkannt wird und Korrekturempfehlungen bekannt sind, dann werden diese Empfehlungen dem Benutzer bereitgestellt, außer dies würde die Sicherheit oder den Zweck des Inhalts gefährden. **AA**
    - **Fehlervermeidung** (rechtliche, finanzielle, Daten): Für Webseiten, die eine für den Benutzer auftretende rechtliche Verpflichtung oder finanzielle Transaktion zur Folge haben, die Benutzer-gesteuerte Daten in Datenspeicherungssystemen ändern oder löschen oder die Testantworten des Benutzers abschicken, gilt mindestens eines der Folgenden: **AA Reversibel**: Versendete Daten sind reversibel. **Geprüft**: Vom Benutzer eingegebene Daten werden auf Eingabefehler überprüft und der Benutzer erhält die Gelegenheit, diese zu korrigieren. **Bestätigt**: Es gibt einen Mechanismus, um Informationen zu überprüfen, zu bestätigen und zu korrigieren, bevor sie endgültig
    - Es gibt eine **kontextsensitive Hilfe**. **AAA**
    - **Fehlervermeidung** (alle): Für Webseiten, die verlangen, dass der Benutzer Informationen absendet, gilt mindestens eines der Folgenden: **AAA - gleich wie oben Rev, Gep, Best**

- **Robust = Inhalte müssen robust genug sein, damit sie zuverlässig von einer großen Auswahl an Benutzeragenten einschließlich assistierender Techniken interpretiert werden können**
    - **Kompatibel** = Maximieren Sie die Kompatibilität mit aktuellen und zukünftigen Benutzeragenten, einschließlich assistierender Techniken.
    - **Syntaxanalyse:** Bei Inhalt, der durch die Benutzung von Auszeichnungssprache implementiert wurde, haben Elemente komplette Start- und End-Tags **A**
    - Name und Rolle durch Software bestimmt werden; Zustände, Eigenschaften und Werte, die vom Benutzer festgelegt werden können

Gegenüberstellung

WCAG1	WCAG 2
• Richtlinien/Guidelines	• Prinzipien/Principles
• Unterteilt in **Prüfpunkte/Checkpoints**	• Unterteilt in Richtlinien/Guidelines
• Prioritäten 1, 2, 3	• **Erfolgskriterien/Success Criteria** mit A, AA, AAA
	• Eine Bedingung kann in mehreren Kriterien mit unterschiedlicher Stufe enthalten sein

Eigener Upgrade Guide von W3C

Wenn WCAG 1 erfüllt → leichter Umstieg

Welchen Technologien wird vertraut (wichtige Basis!)

Tabelle der checkpoint-Vergleiche

W3C schlägt schrittweises Upgraden vor, beginnend bei den häufigst besuchten Seiten

Selten genutzte Seiten können z.B. nie upgegraded werden

WCAG wird in der Struktur einer Site verankert (z.B. Templates, CSS, Scripts, ...)

Wichtig

Autoren entsprechend schulen

Immer wiederkehrende stichprobenartige Überprüfungen

Neue Technologien überprüfen