HTML5 und WAI-ARIA

Oder: Die Suppe nicht versalzen

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Zur Person

Jan Eric Hellbusch

- Freiberuflich tätig – Accessibility-Beratung
- Seit 2000 zahlreiche Veröffentlichungen zur Barrierefreiheit im Web
- Mitglied der Webkrauts für ein besseres Web
- Ehrenamtlich engagiert in der Blinden- und Sehbehindertenselbsthilfe seit 1997 (DVBS, DBSV, PRO RETINA Deutschland)

hellbusch@2bweb.de http://2bweb.de

Früher gab es einfache Kost

Es wird viele Möglichkeiten geben, barrierefreie Webseiten mit HTML5 und WAI-ARIA semantisch anzureichern.

Von strukturierten Dokumenten mit einer Navigation bewegen wir uns in Richtung semantisch angereicherten Anwendungen.

Heute: Barrierefreiheit in Screenreadern

Barrierefreiheit bedeutet definitiv mehr als nur die Zugänglichkeit mit Screenreadern.

In meinem Vortrag geht es aber ausschließlich um die Zugänglichkeit mit Screenreadern.

- Windows-Screenreader: JAWS, Cobra, NVDA, Window Eyes, Supernova u.a.m.
- Apple-Screenreader: VoiceOver.
- Linux-Screenreader: Orca.

hellbusch@2bweb.de http://2bweb.de

Was macht ein Screenreader

Ein Screenreader – so deutet es der Name schon an – ist eine Software, die den Bildschirminhalt vorliest.
Tatsächlich ist die Software wesentlich leistungsfähiger:

- alternative Schnittstelle zum Computer
- Ausgabe der Inhalte, Menüs/Symbolleisten, Dialogfenster u.v.m. in Sprachausgabe oder auf Braille-Zeile
- Erweitertes Bedienkonzept für die Tastatur

hellbusch@2bweb.de http://2bweb.de

Braille-Zeile

Mit einem Screenreader wird u.a. eine Braille-Zeile gesteuert.

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Screenreader, DOM und Accessibility API

Ein Screenreader holt seine Informationen von verschiedenen Stellen im Betriebssystem:

- Accessibility API (Schnittstelle des Betriebssystems), z.B. MSAA, UI Automation und IAccessible2 für Windows, Mac OS X Accessibility Protocol für Apple, Accessibility Toolkit für Linux u.a.
- Document Object Model (DOM)
- Off screen model (OSM)

Tendenz: Weg vom Bildschirm hin zur Accessibility API.

Viele Köche ... und viele Rezepte

Die Barrierefreiheit verantworten viele:

1. Browser, Screenreader (UAAG 1.0) auf verschiedenen Betriebssystemen (Accessibility APIs)
2. Redaktionssysteme, JavaScript-Bibliotheken (ATAG 1.0)
3. Webentwickler, Redakteure (WCAG 2.0)
4. Nutzer (keine Richtlinie)

Je früher Barrierefreiheit gesichert ist, umso besser ist das Ergebnis für den Nutzer.

hellbusch@2bweb.de http://2bweb.de

Rezepte für Webseiten

Die besten Rezepte für barrierefreies Webdesign stecken in HTML5 und WAI-ARIA.

(Es geht nach wie vor nur um Screenreader)

Das liegt u.a. daran, dass aktuelle Browser zunehmend die Accessibility API korrekt füttern.

hellbusch@2bweb.de http://2bweb.de

HTML5

HTML5 ist Work in Progress.

Aber: Ist HTML5 soweit?

Und noch wichtiger: Kann mit HTML5 Barrierefreiheit erreicht werden?

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Ist HTML5 soweit?

Die knappe Antwort:

Noch nicht.

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Ist HTML5 soweit?

Die ausführliche Antwort:

Noch lange nicht.

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Es gibt viele Baustellen

HTML5 bündelt und vereinfacht zahlreiche Techniken, z.B.:

- neue Semantik
- Formulare/User Interface
- Audio/Video
- SVG/Canvas
- JavaScript-Werkzeuge
- WAI-ARIA

Stand: März 2014

Neue und verbesserte Semantik (I)

HTML5 bietet eine Vielzahl von neuen Elementen beispielsweise zur Gliederung einer Seite in Regionen:

- <article> ist für Teilinhalte, die für sich stehen können ("unabhängige Inhalte" wie z.B. ein Blog-Post, ein Kommentar oder ein Zeitungsartikel).
- <aside> ist für Inhalte, die unabhängig vom Inhalt für sich stehen könnten, wie Sidebars.
- <footer> ist für ergänzende Angaben zu einer Region, z.B. Autor oder weiterführende Links zu einem Artikel.

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Neue und verbesserte Semantik (II)

- <header> ist für den Kopfbereich einer Seite, z.B. Logo/Banner, und für den Kopfbereich anderer Regionen.
- <nav> ist für Links, z.B. Navigationsleisten.
- <section> ist für generische Inhalte (z.B. Abschnitt eines längeren Textes). Üblicherweise sollten <article> oder genutzt werden.

HTML 5.1: Sections

Darüber hinaus ist <main> zur Gruppierung des Hauptinhalts einer Webseite vorgesehen.

HTML 5.1: Grouping content

hellbusch@2bweb.de http://2bweb.de

Ersatz für

Die HTML5-Elemente für die Gliederung sowie <main> sind nur bedingt ein Ersatz für

- <aside>, <footer>, <header>, <main>, <nav> und <section> sind tastaturbedienbare Regionen.
- Verschachtelungen der Elemente ist möglich (bis auf <main>), aber oft nicht sinnvoll.
- <section> und <article> sind problematisch im Screenreader

```
<section>
<h1>wird zu H2</h1>
<section><h1>wird zu H3</h1></section>
<section><h2>wird zu H4</h2></section>
</section>
```


hellbusch@2bweb.de http://2bweb.de

Beispiel <figure>

Bekanntes Problem:

- Bilder mit Bildunterschrift erhalten den gleichen Text als Alternativtext und Bildunterschrift.

Mit HTML5 wird es möglich sein, auch nur einen Text zu vergeben:

<figure>
<img src="bild.jpg">
<figcaption>Bildunterschrift **kann** als alt-Text dienen </figcaption>
</figure>

DieGruppierung mit FIGURE ist noch nicht zugänglichkeitsunterstützend.

HTML5: Techniques for providing useful text alternatives

<input>-Elemente

Zu Typen von Eingabefeldern wie „date“, „color“ oder „tel“ heißt es immer: „Nicht alle <input>-Typen werden unterstützt, aber wenn sie in einem Browser nicht unterstützt werden, wird ein normales Eingabefeld angezeigt.“

Aber:

- In IE11 ist nur „range“ zugänglichkeitsunterstützend
- In Firefox 27 kommt etwas mehr bei der Accessibility API an, aber nur „range“ kommt auch beim Screenreader an (JAWS/NVDA auf Windows).

Audio und Video

Früher war barrierefreies Multimedia aufwendig.

Barrierefreie Gestaltung eines Online-Videos

Audio kann zwischenzeitlich (fast) wie folgt eingebunden werden (IE11, keine Tastaturbedienung in FF/Chrome):
<audio src=„audio.mp3" controls ></audio>

Das HTML5 <audio>-Tag - Stand 2014

<video> funktioniert in FF/IE, allerdings keine Audiodeskription, nur Untertitel.

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

<canvas> und SVG

Bei interaktiven Inhalten lässt die Barrierefreiheit auf sich warten.

- <canvas>/JavaScript: IE und Firefox unterstützen die Semantik, Chrome nur die Tastaturbedienung.
- SVG/XML: Nach 15 Jahren sollte mehr in Sachen Barrierefreiheit erwartet werden können.

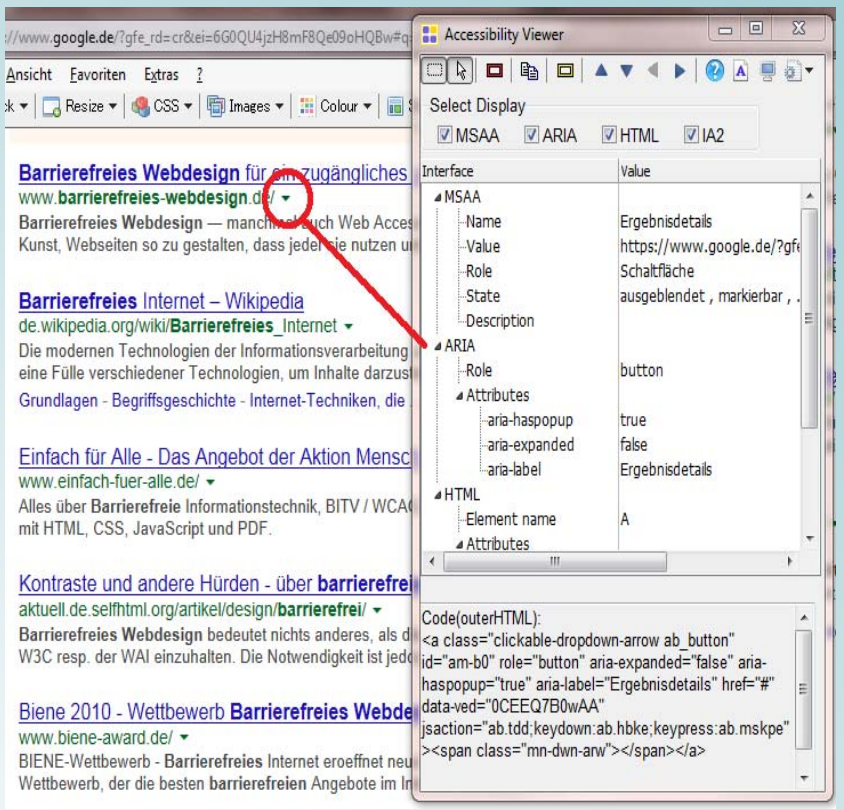
Die Theorie-Phase ist vorbei, aber die Phase der Experimente nicht.

Halbvolle Teller für den Screenreader

Screenreader holen die verfügbaren Informationen von der Accessibility API ab. Infos aus der UAAG:

API Map Master Index

Einsicht mit Aviewer.



© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

WAI-ARIA

Wenn HTML5 noch nicht fertig ist, so funktioniert WAI-ARIA schon.

WAI-ARIA ist ein Satz von Attributen für HTML, SVG usw.:

- Rollen
- Zustände und Eigenschaften
- Tastaturbedienung

WAI-ARIA ist teilweise eine Zwischenlösung.

Accessible Rich Internet Applications (ARIA) 1.0

Rollen

In der WAI-ARIA-Spezifikation werden vier Arten von Rollen definiert. Die Rollen dienen unterschiedlichen Zwecken:

- landmark roles: Mit Screenreadern anspringbare Regionen einer Seite.
- Document structure roles: Semantische Informationen in einer Webseite.
- Widget roles: Semantische Kennzeichnung komplexere, interaktive UI Komponente (zum Beispiel Tab-Panels, Baumstrukturen oder Akkordeons).
- Abstract roles: Sie dienen der Klassifizierung der anderen Rollen und sind für die Webentwicklung völlig uninteressant.

Landmark roles für die Tastaturbedienung

Bei den landmark roles geht es vor allem um ein Navigationskonzept für Tastatur- und insbesondere Screenreader-Nutzer. Die landmark roles dienen der strukturellen Navigation und werden die seit vielen Jahren genutzte Technik der Navigation über Überschriften ersetzen.

Statt

- <h2 class="unsichtbar">Navigation</h2> ... Navigationselemente

werden die Regionen einer Seite mit landmark roles aufgeteilt:

- <nav id="navi" role="navigation">
  ... Navigationselemente
  </nav>

Landmarks mit WAI-ARIA

hellbusch@2bweb.de http://2bweb.de

Landmark roles und HTML5-Elemente

- Application (keine HTML5-Entsprechung)
- Banner (<header role="banner"> ... </header>)
- Complementary (<aside role="complementary"> ... </aside>)
- Contentinfo (<footer role="contentinfo"> ... </footer>)
- Form
- Main (<main role="main"> ... </main>)
- Navigation (<nav role="navigation"> ... </nav>)
- Search (keine HTML5-Entsprechung)

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Landmark roles beschriften

Bei nicht eindeutigen Regionen sollte eine Beschriftung berücksichtigt werden:

- <nav id="navi" role="navigation"
  aria-label="Hauptnavigation">
  ... Navigationselemente
  </nav>

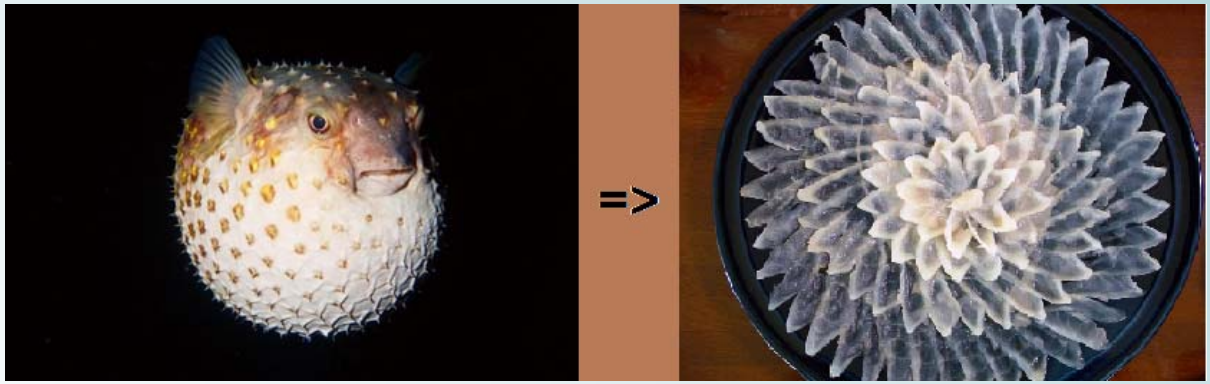
Rückwärtskompatibel:

- <nav id="navi" role="navigation"
  aria-labelledby="naviID">
  <h2 class="unsichtbar" id="naviID">Hauptnavigation</h2>
  ... Navigationselemente
  </nav>

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Vorsicht bei application

Die landmark role application sollten Sie nur einsetzen, wenn Sie genau wissen, was Sie tun.



Die Rolle setzt die Navigation des Screenreaders praktisch außer Kraft. Es muss dann alles mit der Tab-Taste (und vor allem ohne Bildschirm) bedient werden können.

Die meisten Webanwendungen sind im Sinne von WAI-ARIA Dokumente, die in Teilen mit den widget roles abgebildet werden können.

Document structure roles

- article (<article>)
- columnheader (<th>)
- definition (<dfn>, <dt>)
- directory (-)
- document (<body>)
- group (<fieldset>, u.a.)
- heading (<h1> bis <h6>)
- img (<img>)
- list (<ul>, <ol>, <dl>)
- listitem (<li>)
- math (-)
- note (-)
- presentation ()
- region (<section>, <frame>, <iframe>)
- row (<tr>)
- rowgroup (<thead>, <tfoot>, <tbody>)
- rowheader (<th>)
- separator (<hr>)
- toolbar (<menu type=„toolbar“>)

Zur Rolle „presentation“

role="presentation" entfernt "nur" die erforderlichen Tags des Elements. Aus

- <table role="presentation"><tr><td>Ein Text <abbr title="zum Beispiel">z.B.</abbr> über ARIA</td></tr></table>

wird in der Accessibility API zu:

- Ein Text <abbr title="zum Beispiel">z.B.</abbr> über ARIA

Layout-Tabellen sind wie Briefe in Excel

hellbusch@2bweb.de http://2bweb.de

Widget roles (I)

Widget roles identifizieren zunächst interaktive Elemente einer Webseite. Sie müssen oft mit weiteren Attributen ergänzt werden, um Zustände und Eigenschaften der Komponenten korrekt an den Browser vermitteln zu können. Grundsätzlich muss die Tastaturbedienung von der Webentwicklung bereitgestellt werden.

1. Semantik mit role (HTML oder Skripten).
2. Zustände und Eigenschaften im HTML oder mit Skripten.
3. Tastaturbedienung mit Skripten.
4. Aktualisierung der Zustände und Eigenschaften mit Skripten.

Widget roles (II)

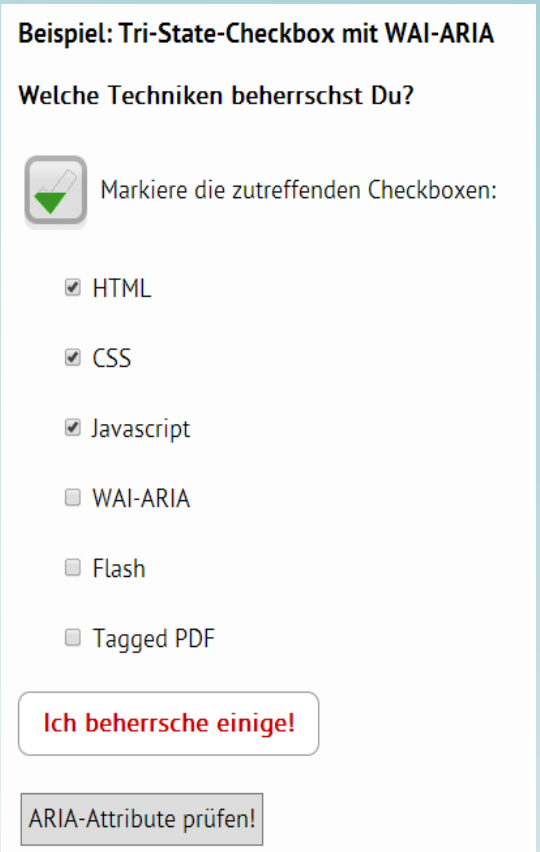
Mit den WAI-ARIA-roles können Inhaltstypen samt diverser Eigenschaften identifiziert werden. Ein kann role="slider" erhalten; die Bedienung mit der Tastatur (z.B. Pfeiltasten) muss der Webentwickler bereitstellen, ebenso die Aktualisierung des Zustands (Wert des Sliders).

Einfaches Beispiel: Tri-State-Checkbox

Wenn für eine Checkbox drei Zustände gebraucht werden, kann eine Grafik oder ein Button die Rolle checkbox erhalten. Darüber hinaus:

1. Tastaturbedienung (tabindex="0" für Grafiken).
2. Zustand mit aria-checked (aktiviert, teilweise aktiviert, nicht aktiviert).
3. Name (aria-label für Beschriftung/label)

Tri-State-Checkbox: Mit Tastatur und Screenreader bedienbar



Das WAI-ARIA-Rezept

Von den Zutaten zur Zubereitung; vier Regeln für WAI-ARIA:

1. HTML: Wenn es mit HTML geht, dann soll HTML genutzt werden.
2. Semantik: WAI-ARIA soll die Semantik von HTML nicht verändern (sondern bei Fehlen hinzufügen).
3. Tastaturbedienung: Alle interaktiven Steuerelemente müssen mit der Tastatur bedient werden können.
4. Sehr wichtig: Halten Sie sich an die erste Regel nur dann nicht, wenn das HTML-Element nicht zugänglichkeitsunterstützend ist oder wenn HTML keine passende Semantik bietet.

hellbusch@2bweb.de http://2bweb.de

Erste Regel für WAI-ARIA

Verwenden sie HTML statt WAI-ARIA:

- ... statt role="link"
- <table>...</table> statt role="grid"
- <img /> statt role="img"

Die Vergabe einer Rolle überschreibt in jedem Fall die HTML-Semantik. Das gilt teilweise auch für Eigenschaften:

- aria-label überschreibt den Text oder den Alternativtext
- aria-checked greift nur bei role="checkbox"

HTML5 wird langsam gar

Bis vor Kurzem war das HTML5-Attribut hidden nicht zugänglichkeitsunterstützend und WAI-ARIA bügelte aus:

- <p hidden aria-hidden="true">Diesen Text kriegt niemand mit.</p>

Jetzt ist hidden zugänglichkeitsunterstützend und das WAI-ARIA-Attribut ist nicht mehr erforderlich:

- <p hidden>Diesen Text kriegt niemand mit.</p>

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Zweite Regel: Bitte kein Pfusch

Statt

- Meine Überschrift

bitte folgendes verwenden:

- <h3>Meine Überschrift</h3>

Siehe auch erste Regel. Entwickler sollten die richtigen Elemente nutzen bevor WAI-ARIA hinzugefügt wird.

hellbusch@2bweb.de http://2bweb.de

Wenn die Semantik im HTML fehlt

Die Rollen und Eigenschaften sind dann wichtig, wenn die Semantik fehlt. Aus:

-

kann WAI-ARIA etwas sinnvolles für Screenreader machen:

-

Die Rollen und Eigenschaften sind auch dann wichtig, wenn die Semantik in HTML5 noch nicht oder noch nicht gut unterstützt wird:

- <footer role=“contentinfo“ aria-label=“Zum Autor des Artikels”> ... </footer>

hellbusch@2bweb.de http://2bweb.de

Kritischer ist die Verhaltensebene

Wenn eine Überschrift anklickbar sein soll, dann bitte nicht:

- <h3 role="link" onclick="machwas();">Mein anklickbarer Überschriftentext</h3>

Sondern

- <h3>Mein anklickbarer Überschriftentext</h3>

Grund: Wenn die Überschrift die Rolle eines Links bekommt, ist sie keine Überschrift mehr, sie kann mit der strukturellen Navigation des Screenreaders nicht angesprungen werden und außerdem ist der Link ohne ein tabindex mit der Tastatur nicht fokussierbar und somit auch nicht bedienbar.

Umgekehrt genauso

Umgekehrt bleibt ein ursprünglich tastaturbedienbares Element auch nach einer Neudefinition mit role mit der Tastatur bedienbar.

- Fokussierbare Überschrift

Dritte Regel: Auf die Tastaturbedienung kommt es an

Das tabindex-Attribut kann mit dem Wert 0 genutzt werden. Das ist sinnvoll, wenn UI Komponente nicht mit interaktiven Elementen wie Links oder Steuerelemente gestaltet werden:

- Schließen
- Für komplexere UI Komponente muss die Tastatursteuerung per JavaScript hinzugefügt werden. Die Tastenschläge sind nicht genormt, aber sollten sich an Konventionen für das Betriebssystem orientieren. Siehe hierzu die "Design Patterns" in:

  WAI-ARIA 1.0 Authoring Practices

4. Regel: Wann WAI-ARIA sein muss

WAI-ARIA geht über HTML5 hinaus

WAI-ARIA bietet eine Reihe sogenannter "widget roles" für solche UI-Elemente, die in HTML5 nicht vorgesehen sind, z.B. die Tri-State-Checkbox.

Es gibt viele weitere UI-Komponenten, wofür WAI-ARIA essentiell ist, z.B.: Combo-Boxen, Baumnavigationen, Reiternavigationen, Modalfenster oder Akkordeons.

Vorher muss die dritte Regel verinnerlicht werden.

hellbusch@2bweb.de http://2bweb.de

Zusammenrühren

Vorab: WAI-ARIA ist nicht nur für HTML vorgesehen; sie wird in anderen Markup-Sprachen ebenfalls benötigt.

- WAI-ARIA soll zusammen mit HTML5 genutzt werden. Da HTML5 zunehmend zugänglichkeitsunterstützend wird, kann WAI-ARIA zunehmend weggelassen werden.
- HTML5-Semantik wird nach und nach in den Browsern implementiert, aber nicht alle Informationen werden an die Accessibility API weitergeleitet.
- Wo HTML5 keine Mittel bietet, muss WAI-ARIA eingesetzt werden.
- Bei altbekannten (HTML 4.01-)Elementen mit Semantik ist die Zuweisung von Rollen nicht erforderlich.
- Bei neuen (HTML5-)Elementen mit Semantik sollten Rollen zugewiesen werden: <nav role="navigation"> ... </nav>.

Die Suppe nicht versalzen

Im Eifer des Gefechts kann WAI-ARIA falsch eingesetzt werden. Beispielsweise können Rollen und HTML5-Attribute unterschiedliche Bedeutung haben:

- <form role=”search”>
  <label for=”query”>Suche nach:</label>
  <input type=”search” id=”query” name=”query”> ...
  </form>

Falsch ist hingegen:

- <input type=”search” role=”search” />

Andere Beispiele:

- <nav> <ul role="navigation"> ... </nav>
- <body role=“application”> … </body>

Schlussfolgerungen

IN HTML5 muss WAI-ARIA mit unterschiedlicher Intensität genutzt werden

- Konzentrieren Sie sich auf semantisch bedeutsames HTML.
- Die landmark roles (einschließlich Beschriftung) sollten als Ersatz für die herkömmliche strukturelle Navigation eingesetzt werden.
- Die document structure roles werden in HTML5 weniger gebraucht.
- Die widget roles werden in den nächsten Jahren wichtig bleiben. Die Bedienbarkeit und die erforderlichen Aktualisierungen der Zustände und Eigenschaften muss in den Bibliotheken gut verankert werden.

Fertig

Vielen Dank.

Haben Sie noch Fragen?

Feedback und Download der Präsentation auf www.hellbusch.de.

© Jan Eric Hellbusch (1967 - 2014)
hellbusch@2bweb.de http://2bweb.de

Bildquellen

Seite 3: pixelio.de - Robby Enders und Wolfgang Dirscherl

Seite 11: pixelio.de - Claudia Hautumm

Seite 12: pixelio.de – Jörg Henkel Hamburg

Seite 27: pixelio.de – Niki Vogt

Seite 27: wikipedia.de – Suguri F

Seite 39: www.gekreuzsiegt.de