

Information und ihre Darstellung: XHTML & CSS


PL Speyer

Daniel Jonietz

2011

XHTML – Extensible Hypertext Markup Language

Unter besonderer
Berücksichtigung von
XHTML 1.1.



Was ist XHTML?

- Textbasierte Auszeichnungssprache zur Auszeichnung von Inhalten in Dokumenten
- Neuformulierung von HTML, (war definiert in SGML (Standard Generalized ML)) in XML (Extensible ML)
- Ziele: u.a. Erweiterbarkeit des Sprachraumes
- XHTML-Dokumente entsprechen den Syntaxregeln von XML

XHTML ist einfacher ...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">  
<>  
<title//  
<p ltr<span></span</p>  
</>
```



XHTML einfach gesagt: Was ist anders (als früher)?

- alle Tags klein schreiben
- alle Tags schließen
- alle Werte in doppelte Anführungszeichen
- gibt keine Attributminimierung

- gibt keine Elemente zur grafischen Auszeichnung

XHTML-Varianten und Versionen

- XHTML 1.0 entspricht HTML 4.01
 - transitional, frameset, strict
- XHTML 1.1 entspricht im Wesentlichen 1.0 strict
 - verzichtet auf weitere grafische Elemente (font)
- Noch Zukunft:
XHTML 2.0 mit grundlegenden Änderungen:
 - "XHTML 2 wurde so definiert, wie eine Markup-Sprache sein sollte – nicht, wie Markup-Sprachen heutzutage verwendet werden."

Grundgerüst Version 0

```
<html>  
<head>  
<title>Grundgerüst</title>  
</head>  
<body>  
</body>  
</html>
```

Wichtige Tags zur Auszeichnung

- Überschrift: `<h1>...</h1>` bis `<h6>...</h6>`
- Absatz: `<p>...</p>`
- Listen:
 - Ungeordnet: `...`
 - Geordnet: `...`
 - Listeneintrag: `...`

Steckbrief

Steckbrief

von Daniel Jonietz

- ◆ Beruf: Lehrer am Burggymnasium
- ◆ Fächer: Informatik, Mathematik
- ◆ Geburtsdatum: 23. Juni 1976
- ◆ Kontakt:
 - ◇ E-Mail: daniel@jonietz.de
 - ◇ Jabber: [daniel@jabber.jonietz.de](jabber: daniel@jabber.jonietz.de)

```
<h1>Steckbrief</h1>
<h2>von Daniel Jonietz</h2>
<ul>
  <li>Beruf: Lehrer am Burggymnasium</li>
  <li>Fächer: Informatik, Mathematik</li>
  <li>Geburtsdatum: 23. Juni 1976</li>
  <li>Kontakt:
    <ul>
      <li>E-Mail: daniel@jonietz.de</li>
      <li>Jabber: daniel@jabber.jonietz.de</li>
    </ul>
  </li>
</ul>
```

Verweise

```
<a href="...">...</a>
```

Steckbrief

von Daniel Jonietz

- Beruf: Lehrer am [Burggymnasium](#)
- Fächer: Informatik, Mathematik
- Geburtsdatum: 23. Juni 1976
- Kontakt:
 - ◊ E-Mail: daniel@jonietz.de
 - ◊ Jabber: [daniel@jabber.jonietz.de](jabber:jonietz.de)

Bilder

- Müssen in einem Block-Element stehen, d. h. zum Beispiel in einem Absatz
- Müssen einen Alternativ-Text haben
- Werden mit "inhaltsleeren" Tags geschrieben

```

```

Darstellung von Information

- Trennung von Inhalt und Layout
 - im ersten Schritt: auf jegliches Layout verzichten
 - im zweiten Schritt: Inhalte "layouten"
- bedeutungsvolles, am Inhalt orientiertes Markup
 - semantischen Code schreiben
- Nebenaspekte
 - Codierungsfragen z.B. für Farben: #99cc99
 - nicht mehr so relevant: HTML-Entities: ü

Der Inhalt: Semantisches HTML

`<body> . . . </body>`

- „Semantischer Code ist die Verwendung von HTML-Elementen, welche in ihrem Kontext das Höchstmaß an Bedeutung kommunizieren.
- Semantischer Code ist Code, der einen Bezug zum Inhalt herstellt. Er umschließt oder repräsentiert Inhalt und übermittelt dessen Bedeutung oder Funktion.
- Semantischer Code ist nicht präsentationsgebunden, sondern informationsgetrieben. Er transportiert Sinngehalt nicht über die visuelle Gestalt, sondern über universell verständliche “Etiketten”. Er ist eine plattformunabhängige Meta-Information.
- Semantischer Code kommuniziert nicht nur Bedeutung, sondern verleiht auch Struktur. Es werden Beziehungen hergestellt und Hierarchien aufgebaut.“

Layout und inhaltliche Bedeutung

- Er war spät. Sie war froh, *dass* er kam.
 - Betonung: `...`
- Die *Times* schreibt ...
 - Quellenangabe, Zitat: `<cite lang="en">...</cite>`
- Ein Redner *par excellence*.
 - Sprachwechsel: `...`

XHTML als formale Sprache

- XHTML ist eine formale Sprache
Grammatik besteht aus:
 - syntaktischen Regeln von XML
 - Elementstrukturen, die die Dokumententyp-Definition (DTD) vorgibt (also welche Elemente gibt es, wie dürfen sie verschachtelt werden, welche Attribute gibt es)
- Validatoren können Verstöße gegen Syntax oder Struktur feststellen

Grundgerüst Version 1

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xml:lang="de">
<head><title>Erweitertes Grundgerüst</head>
</head>
<body>
</body>
</html>
```


wohlgeformte XML-Dokumente

- Es existiert genau ein Wurzelement (html)
- Alle Elemente werden durch Start- und Endtag begrenzt oder bestehen aus einem Leeres-Element-Tag
- Elemente sind korrekt verschachtelt
- Attributwerte stehen in Anführungszeichen
- Es gibt keine Attributminimierung

Wohlgeformtheit und Validität

- Dokumente müssen wohlgeformt sein, damit sich prüfen lässt, ob sie valide sind.

Valide Dokumente

- Ein Dokument heißt valide (oder gültig), wenn es erfolgreich gegen die zu Grunde liegende DTD geprüft werden kann.
- Elemente, Attribute und Attributwerte dürfen nur so verwendet und verschachtelt werden, wie es die gewählte Sprache erlaubt



Exemplarische Bestandsaufnahme

- <http://www.rlp.de>
 - 151 Fehler im HTML 4.01
 - 5 Fehler, 58 Warnungen im CSS
 - Designministerium: 14/100 Punkte
 - Zugänglichkeit: Tests nach W3C WCAG

	Automatic Checkpoints			Manual Checkpoints		
	Status	Errors	Instances	Status	Warnings	Instances
Priority 1	⊗	4	5	⚠	12	750
Priority 2	⊗	3	613	⚠	21	793
Priority 3	⊗	2	82	⚠	10	10

Warum fehlerfreie Webseiten?

- Fehlerbehandlung ist nicht genormt
- Es gibt keine Garantie, dass Browser syntaktisch falsche Dokumente in Zukunft noch anzeigen
- Als `application/xhtml+xml` ausgelieferte XHTML-Webseiten werden heute schon nur angezeigt, wenn das Dokument keine Fehler enthält

Validität ist nicht alles!

- "Der Satz:
'Der Mann, der die Brücke, die über den Fluss, der im Winter Hochwasser, das ganz schlammig ist, führt, führt, baute, lebt nicht mehr'
ist eine valide formulierte Aussage. Dennoch ist es unsinnig, solche Sätze zu formulieren."
Steffi Abel, nach c't 12/2005

Ausblick: Datentypen

- In DTDs lassen sich keine Datentypen definieren, es gibt nur den universellen Datentyp CDATA (Zeichenfolge)
- Es gibt Beschreibungen von XHTML als XML Schema, die Struktur und Daten genauer beschreiben - inklusive Datentypen und Wertebereichsbeschränkungen

CSS – Cascading Style Sheets

CSS 2



Verknüpfung mit einem Stylesheet

- Der Kopf der XHTML-Datei muss ergänzt werden:

```
<head>  
  <title>Steckbrief von Daniel Jonietz</title>  
  <link rel="stylesheet" type="text/css"  
    href="steckbrief.css" media="screen" />  
</head>
```

Die CSS-Datei

```
body
{
  background-color: #C8D2F0;
  font-size: 1em;
  font-family: "Trebuchet MS", Verdana, sans-serif;
  color: black;
}
```

Vorher - nachher

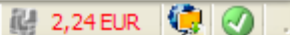
Steckbrief

von Daniel Jonietz



- ◆ Beruf: Lehrer am [Burggymnasium](#)
- ◆ Fächer: Informatik, Mathematik
- ◆ Geburtsdatum: 23. Juni 1976
- ◆ Kontakt:
 - ◊ E-Mail: daniel@jonietz.de
 - ◊ Jabber: [daniel@jabber.jonietz.de](jabber:daniel@jabber.jonietz.de)

Fertig



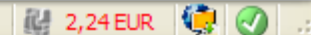
Steckbrief

von Daniel Jonietz



- ◆ Beruf: Lehrer am [Burggymnasium](#)
- ◆ Fächer: Informatik, Mathematik
- ◆ Geburtsdatum: 23. Juni 1976
- ◆ Kontakt:
 - ◊ E-Mail: daniel@jonietz.de
 - ◊ Jabber: [daniel@jabber.jonietz.de](jabber:daniel@jabber.jonietz.de)

Fertig



Die Überschriften

```
h1, h2
```

```
{
```

```
  font-weight: bold;
```

```
  font-family: Georgia, "Trebuchet MS", sans-serif;
```

```
  color: #9160a1;
```

```
}
```

Steckbrief

von Daniel Jonietz

```
h1
```

```
{
```

```
  font-size: 200%;
```

```
}
```

Steckbrief

von Daniel Jonietz

Die Links

```
a:link { text-decoration: none; color: #466498; }
```

```
a:visited  
{  
  text-decoration: none;  
  color: #466498;  
}
```

```
a:hover  
{  
  text-decoration: underline;  
}
```

- Beruf: Lehrer am [Burggymnasium](#)
- Fächer: Informatik, Mathematik
- Geburtsdatum: 23. Juni 1976
- Kontakt:
 - E-Mail: daniel@jonietz.de
 - Jabber: [daniel@jabber.jonietz.de](jabber:jonietz.de)

- Beruf: Lehrer am [Burggymnasium](#)
- Fächer: Informatik, Mathematik
- Geburtsdatum: 23. Juni 1976
- Kontakt:
 - E-Mail: daniel@jonietz.de
 - Jabber: [daniel@jabber.jonietz.de](jabber:jonietz.de)

Die Aufzählung

- Beruf: Lehrer am Burggymnasium
- Fächer: Informatik, Mathematik
- Geburtsdatum: 23. Juni 1976
- Kontakt:
 - E-Mail: daniel@jonietz.de
 - Jabber: [daniel@jabber.jonietz.de](jabber:daniel@jabber.jonietz.de)

```
ul
{
  list-style-image: url(square.png);
}
```

```
ul ul
{
  list-style-image: url(minisquare.png);
  padding-left: 24px;
}
```

- Beruf: Lehrer am Burggymnasium
- Fächer: Informatik, Mathematik
- Geburtsdatum: 23. Juni 1976
- Kontakt:
 - E-Mail: daniel@jonietz.de
 - Jabber: [daniel@jabber.jonietz.de](jabber:daniel@jabber.jonietz.de)

Das Bild

```
img
{
  float: right;
}
```



Mit Blick auf Barrierefreiheit...

- Hervorhebungen
`...` und `...`
- Adressen
`<address>...</address>`
- Abkürzungen
`<abbr>...</abbr>`
- Akronyme
`<acronym title="Institut für schulische Fortbildung und schulpsychologische Beratung">IFB</acronym>`

Verweise klassifizieren

- Seite enthält verschiedene Arten von Links
 - auf andere Webseite
 - auf E-Mail-Adresse
 - auf XMPP-Jabber-Adresse
- Alles sind aber normale `<a>`-Tags, so dass eine Unterscheidung im CSS allein über den Tag-Typ nicht geht => Zugriff auf Attribute

Änderungen am XHTML-Code

```
<li>Beruf: Lehrer am <a class="extern"
  href="http://burg-kl.de">Burggymnasium</a></li>
<li>Fächer: Informatik, Mathematik</li>
<li>Geburtsdatum: 23. Juni 1976</li>
<li>Kontakt:
  <ul>
    <li>E-Mail: <a class="email"
      href="mailto:daniel@jonietz.de">daniel@jonietz.de</a
    ></li>
    <li>Jabber: <a
      ref="xmpp:daniel@jabber.jonietz.de">daniel@jabber.jo
      nietz.de</a></li>
  </ul>
</li>
```

Auszeichnung der Klassen im CSS



```
a.extern  
{  
  text-decoration: underline;  
}
```

```
a.email  
{  
  font-style: italic;  
}
```

- Beruf: Lehrer am [Burggymnasium](#)
- Fächer: Informatik, Mathematik
- Geburtsdatum: 23. Juni 1976
- Kontakt:
 - E-Mail: daniel@jonietz.de
 - Jabber: daniel@jabber.jonietz.de



Oder sogar...

- Beruf: Lehrer am »Burggymnasium
- Fächer: Informatik, Mathematik
- Geburtsdatum: 23. Juni 1976
- Kontakt:
 - E-Mail:  daniel@jonietz.de
 - Jabber:  daniel@jabber.jonietz.de



```
a.extern {  
    background-image: url(extern.png);  
    background-position: top left;  
    background-repeat: no-repeat;  
    padding-left: 9px;  
}
```




```
a.email {  
    background-image: url(mail.png);  
    background-position: top left;  
    background-repeat: no-repeat;  
    padding-left: 18px;  
}
```

Individualformate

- Ein einzelnes Element soll angesprochen werden?
- Jedes einzelne HTML-Tag kann mit einer ID versehen und über dieses individuell angesprochen werden:
`id="..."`
Zugriff im CSS: #

Viele Adressen...

■ Kontakt:

- E-Mail:  daniel@jonietz.de
- E-Mail:  jonietz@burg-kl.de
- E-Mail:  jonietz@mnu2008-kl.de
- Jabber:  [daniel@jabber.jonietz.de](xmpp:daniel@jabber.jonietz.de)

```
<li>E-Mail: <a class="email" id="bevorzugte_adresse"
  href="mailto:daniel@jonietz.de">daniel@jonietz.de</a></li>
<li>E-Mail: <a class="email" href="mailto:jonietz@burg-
  kl.de">jonietz@burg-kl.de</a></li>
<li>E-Mail: <a class="email" href="mailto:jonietz@mnu2008-
  kl.de">jonietz@mnu2008-kl.de</a></li>
<li>Jabber: <a class="jabber"
  href="xmpp:daniel@jabber.jonietz.de">daniel@jabber.jonietz.de</a></li>
```

```
#bevorzugte_adresse {
  font-style: italic;
}
```

div und span

- Mit den Tags `<div>` und `` können Elemente zusammengefasst oder ausgezeichnet werden
- auch für Gestaltung mit CSS, aber nicht nur!
 - `<div>...</div>`
Blockelement (beginnt in einer neuen Zeile)
 - `...`
Inlinebereich
z.B.: Er hat keine `Firewall`