WebTech (2)

Prof. Dr.-Ing. S. Gössner

University of Applied Sciences Lippe & Höxter

Inhalt

- zentrale Einbindung in HTML
- direkte Einbindung in HTML Elemente
- CSS Syntax
- CSS Selektoren
- CSS Eigenschaften

XML - Definition

XML ist eine standardisierte, erweiterbare Auszeichnungssprache zur Erstellung maschinen- und menschenlesbarer Dokumente.

XML wird als *universelles Datenformat* bezeichnet, das insbesondere nicht auf Webdokumente beschränkt ist.

Historie

- Entwurf durch eine elfköpfige Arbeitsgruppe (Vorsitz: James Clark) von Mitte 1996 bis Februar 1998.
- Vorschub durch die Popularität von HTML.
- Untermenge der SGML.
  - SGML litt unter mangelnder Akzeptanz.
  - XML deckt 80% möglicher SGML-Anwendungen mit 20% des SGML-Sprachumfangs ab (*Pareto-Verteilung*).
- W3C Empfehlung XML 1.0 seit Feb. 1998.

Was ist XML

XML ist ...

- zentraler Baustein der Webtechnologien.
- Text, aber nicht primär für menschliche Leser.
- lediglich ähnlich zu HTML.
- ein Format, um strukturierte oder halbstrukturierte Daten abzubilden.
- eine *Metasprache* und damit nicht gebrauchsfertig.
- gleichzeitig eine Familie von Technologien.
- lizenzfrei, plattformunabhängig und breit unterstützt.

Einführendes Beispiel

Ein XML-Dokument wird üblicherweise von einer Software(komponente) verarbeitet. Diese bezeichnet man üblicherweise als den *XML-Parser*.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Recipe name="bread" prep_time="5 mins" cook_time="3 hours">
  <title>Basic bread</title>
  <ingredient amount="3" unit="cups">Flour</ingredient>
  <ingredient amount="0.25" unit="ounce">Yeast</ingredient>
  <ingredient amount="1.5" unit="cups">Warm Water</ingredient>
  <ingredient amount="1" unit="teaspoon">Salt</ingredient>
  <Instructions>
    <step>Mix all ingredients together, and knead thoroughly.</step>
    <step>Cover with a cloth, and leave for one hour in warm room.</step>
    <step>Knead again, place in a tin, and then bake in the oven.</step>
  </Instructions>
</Recipe>
```

[Quelle: Wikipedia]

XML-Deklaration

Muss immer am Anfang eines XML-Dokuments stehen

```
<?xml version="1.0"
     [encoding="{Zeichensatz}"]
     [standalone="{yes|no}"] ?>
```

version
: im Normalfall "1.0" (obligatorisch)

encoding
: verwendeter Zeichensatz, UTF-8 voreingestellt, z.B. ISO-8859-1, westeuropäischer Zeichensatz (optional)

standalone
: yes: keine/interne DTD verfügbar
: no: externe DTD notwendig
: (optional)

Verarbeitungsanweisung

Verarbeitungsanweisungen *(Processing Instructions)* sind Direktiven an den XML-Parser, spezielle Verarbeitungsschritte an bestimmte Programme/-komponenten zu delegieren.

Syntax

```
<?pi-name [attribute1] ... [attributeN]?>
```

Beispiele

```
<?xml-stylesheet type="text/css" href="print.css"?>

<?xml-stylesheet type="text/xsl" href="sample.xsl" ?>

<?xul-overlay href="chrome://test/content/myOverlayxul"?>
```

Obwohl die *XML-Deklaration* dieselbe Syntax besitzt, ist diese **keine** Verarbeitungsanweisung.

Kommentare

Kommentare sind Notizen für den Autor des XML-Dokuments oder für andere menschliche Leser (in Ausnahmefällen für verarbeitende Software), die vom XML-Parser überlesen wird.

Syntax

```
<!-- Kommentar Text -->
```

CDATA - Anweisung

CDATA-Anweisungen werden vom XML-Parser als nicht weiter zu untersuchender Block von Zeichendaten (*character data*) gehandhabt.

Syntax

```
<![CDATA[ Zeichenfolge ]]>
```

Beispiel

```
<raw-data>
  <![CDATA[ a > b && c < d ]]>
</raw-data>
```


Spezielle Zeichen

Gewisse Zeichen, die in XML-Dokumenten eine besondere Bedeutung haben, müssen etwas anders geschrieben werden:

Zeichen	Notation in XML
<	<
>	>
&	&
"	"
'	'

Elemente

Ein XML-Dokument besteht aus genau **einem** Wurzelelement (*root element*).

Syntax

```
<element [attr1="value1"] ... [attrN="valueN"]>
  content
</element>
```

XML-Elemente bestehen aus …

- einem Namen, Gross-/Kleinschreibung ist signifikant in der Schreibweise /[a-z,A-Z,_][a-z,A-Z,_,-,.,0-9]*/
- einem Anfangs-Tag (*start-tag*)
- einem Ende-Tag (*end-tag*)
- Attributen (*optional*)
- einem Inhalt

Elementinhalt

Der Inhalt von Elementen kann …

- aus einfachem Text bestehen (unstrukturiertes Element)
- weitere Elemente enthalten (strukturiertes Element)
- gleichzeitig Elemente und Text enthalten (semistrukturiertes Element)
- leer sein (leeres Element)

unstrukturiertes Element

Syntax: `<Name [Attribut]*>PCDATA</Name>`

Beispiel

```
<paragraph id="p5">
  some text. some text. some text. some text. some text.
  some text. some text. some text. some text. some text.
</paragraph>
```


strukturiertes Element

Syntax: `<Name [Attribut]*>Element+</Name>`

Beispiel

```
<section level="2">
  <subsection> some text. some text. </subsection>
  <subsection> some other text. </subsection>
</section>
```

semistrukturiertes Element

Syntax: `<Name [Attribut]*>[PCDATA|Element]+</Name>`

Beispiel

```
<paragraph>
  some text. some text. some text. some text. some text.
  some text. some text. some text. some text. some text.<section>
  <section>
    <subsection> inner text. inner text. </subsection>
    <subsection> more inner text. </subsection>
  </section>
  some other text. some other text. some other text.
</paragraph>
```


leeres Element

Syntax: `<Name [Attribut]* />`

Beispiel

```
<br/>
<image src="smiley.png" />
<date y="2005" m="feb" d="23" />
```

Attribute

- Ein Element kann eine beliebige Anzahl von Attributen haben.
- Attribute sind Name-Wert-Paare der Form `name="wert"` oder `name='wert'`.
- der Attributname gehorcht derselben Schreibweise wie ein Elementname: /[a-z,A-Z,_][a-z,A-Z,_,-,.,0-9]*/
- Gleichnamige Attribute innerhalb eines Elements sind nicht erlaubt.
- Der Wert eines Attributes ist immer vom Typ *String* (unstrukturierter Inhalt).
- Anführungszeichen innerhalb eines Attributwertes können mittels eines voranstehenden Backslash verwendet werden. *Beispiel:* `winkel="12°45\'23\""`
- Beachte: Die Reihenfolge der Attribute ist belanglos.

Element oder Attribut

Wann also Attribute und wann Elemente verwenden?

```
<student name="smith" firstname="eric" matnr="123456789" />
```

oder

```
<student>
  <name>smith</name>
  <firstname>eric</firstname>
  <matnr>123456789</matnr>
</student>
```

Element oder Attribut (2)

- Ein Attribut kann nur einen String als Wert haben. Bei strukturierten Daten innerhalb eines Elements bieten sich also Unterelemente an.
- Ein Element kann keine gleichnamigen Attribute haben. Bei mehreren gleichartigen Daten bieten sich ebenfalls Unterelemente an.
- Die Reihenfolge der Attribute ist unerheblich, diejenige von Elementen nicht notwendigerweise.
- Einheitliche Darstellung mit Elementen ist oft eleganter, Darstellung mit Attributen kompakter.
- Fazit: Attribute eigenen sich besonders für einfache, unstrukturierte Zusatzinformationen (Metadaten) eines Elements.

Regeln für wohlgeformte Elemente

Ein nichtvalidierender Parser prüft ein XML-Dokument auf *Wohlgeformtheit* (*wellformed document*).

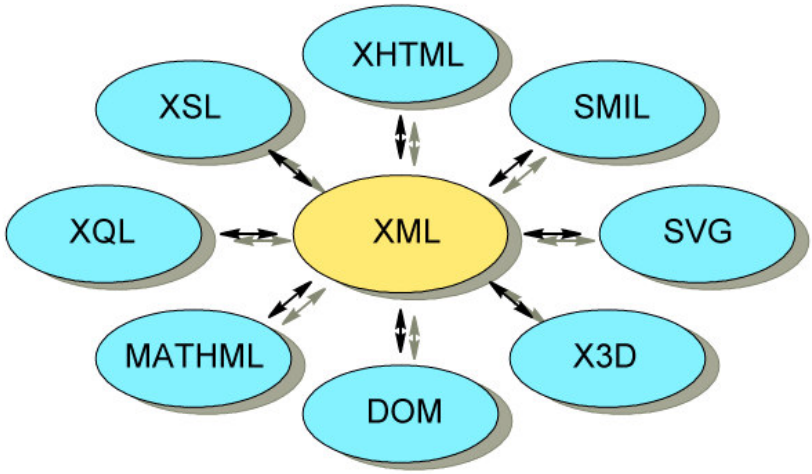
1. Jedes Anfangs-Tag muss ein zugehöriges Ende-Tag haben.
2. Elemente dürfen sich nicht überlappen.
3. XML-Dokumente haben genau ein Wurzel-Element.
4. Elementnamen müssen bestimmten Namenskonventionen entsprechen.
5. XML beachtet grundsätzlich Groß- und Kleinschreibung.
6. XML belässt Formatierungen (white space) im Text.
7. Ein Element darf niemals zwei Attribute mit demselben Namen haben.

Namenskonventionen

- Namen beginnen entweder mit einem Buchstaben oder _: z.B. `first`, `First` oder `_First`
- Nach dem ersten Zeichen sind zusätzlich Zahlen sowie `-` und `.` erlaubt: z.B. `_1st-name` oder `_1st.name`.
- Namen enthalten keine Leerzeichen.
- Namen enthalten kein `:`.
- Namen beginnen nicht mit `xml`, unabhängig davon, ob die einzelnen Buchstaben groß- oder kleingeschrieben sind.
- Diese Konventionen gelten für alle Bezeichner in XML, nicht nur für Element-Namen.

XML - Familie

XML ist die Mutter einer Familie standardisierter Markup Sprachen.



XHTML

XHTML ist eine Neuformulierung von HTML auf der Basis von XML.

Vorteile

- XHTML kann mit anderen XML-basierten Sprachen innerhalb eines Dokuments verbunden werden (*compound document*).
- XHTML kann mit Stylesheet Sprachen (XSLT) weiterverarbeitet werden.
- Auf XHTML kann die standardisierte Programmierschnittstelle DOM angewendet werden.

XHTML - Dokument

```
<?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">
  <head>
    <title>Titel</title>
    <meta name="herkunft" content="fh lippe & höxter"/>
    <link rel="stylesheet" type="text/css"
          href="/css/main.css" media="all" />
  </head>
  <body>
    Inhalt
  </body>
</html>
```

XHTML vs. HTML

- Ein XHTML-Dokument enthält eine XML-Deklaration *(optional)*.
- XHTML Wurzelelement mit Namensraumangabe.
- strenge Überprüfung des korrekten Grundgerüsts.
- XHTML-Elementnamen werden alle *klein* geschrieben (`<body/>` statt `<BODY>`).
- jedes Anfangs-Tag benötigt ein Ende-Tag.
- leere Elemente korrekt schreiben (`<br/>` statt `<br>`).
- Attributwerte immer in Anführungszeichen (`<br/>` statt `<br>`).
- Attribute müssen immer einen Wert besitzen `checked="checked"` statt `checked`).
- `id`-Attribut statt des `name`-Attributes.

Was ist CSS

CSS ist …

- Cascading Style Sheets
- ähnlich den Formatvorlagen eines Textvararbeitungsdokuments.
- eine Sammlung von Vorschriften zur Festlegung des Aussehens von XML/XHTML Elementen.
- eine Steuereinheit für Seitenlayout, Textfluss und Print-Layout.
- ein Hilfsmittel für Webseitengestaltung und Corporate Design.

Einbindung als externe Datei

Syntax

```
<link rel="stylesheet" media="{target}" href="{url}" />
```

Beispiel

```
<html>
  <head>
    <title>Webdokument</title>
    <link rel="stylesheet" media="screen" href="seite.css" />
    <link rel="stylesheet" media="print" href="druck.css" />
    <link rel="stylesheet" media="projection" href="slides.css" />
  </head>
  <body>
  </body>
</html>
```


zentrale Einbindung in HTML

Syntax

```
<style type="text/css">
<![CDATA[
   /* Formatinformation */
]]>
```

Beispiel

```
<html>
  <head>
    <title>Webdokument</title>
    <style type="text/css">
    <![CDATA[
      @media print { /* Drucken ... */
        /* ... */
      }
      @media screen { /* Bildschirm */
        /* ... */
      }
   ]]>
   </style>
 </head>
<body>
</body>
</html>
```

direkte Einbindung in HTML Elemente

Syntax

```
<Name style=" /* ... */ "> [Inhalt]</Name>
```

Beispiel

```
<html>
  <head>
    <title>Webdokument</title>
  </head>
  <body style="/* ... */">
    <h1 style="/* ... */">...</h1>
  </body>
</html>
```


CSS Syntax

```
Selector {property1: value1; ...; propertyN: valueN; }
```

Selector
: Auswahl der gewünschten Elemente (body, h1, table, …).

property
: festzulegende Eigenschaft (color, border-style, …)

value
: Eigenschaftswert (red, solid, …)

CSS Selektoren

Pattern	Meaning
*	Matches any element.
E	Matches any E element (i.e., an element of type E).
E F	Matches any F element that is a descendant of an E element.
E > F	Matches any F element that is a child of an element E.
E:first-child	Matches element E when E is the first child of its parent.
E:link E:visited	Matches element E if E is the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited).
E:active E:hover E:focus	Matches E during certain user actions.
E:lang(c)	Matches element of type E if it is in (human) language c (the document language specifies how language is determined).
E + F	Matches any F element immediately preceded by an element E.
E[foo]	Matches any E element with the "foo" attribute set (whatever the value).
E[foo="warning"]	Matches any E element whose "foo" attribute value is exactly equal to "warning".
E[foo~="warning"]	Matches any E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "warning".
E[lang|="en"]	Matches any E element whose "lang" attribute has a hyphen-separated list of values beginning (from the left) with "en".

DIV.warning	HTML only. The same as DIV[class~="warning"].
E#myid	Matches any E element ID equal to "myid".

[Quelle: W3C]

CSS Eigenschaften

SelfHTML bietet eine gute Übersicht über alle CSS 2.1 Eigenschaften.

CSS Kurzreferenz