

Digitale Medien

Übung

Heute

- Document Type Definition (DTD)
- Formatierung von XML-Dokumenten

eXtensible Markup Language (XML)

Metasprache für hierarchisch strukturierte Textdaten.

Varianten von XML sind z.B.

- XHTML (Webseiten),
- SVG (Vektorgrafik),
- SMIL (Animationen),
- X3D (3D-Szenen),
- RSS (Webfeeds),
- etc.

Vorteile u.a.

- von Menschen und Maschinen lesbar
- Trennung von Inhalt und Präsentation
- beliebig erweiterbar

```
<?xml version="1.0" e
<quiz>
  <question>
    Who was the forty-second
    president of the U.S.A.?
  </question>
  <answer>
    William Jefferson Clinton
  </answer>
  <!-- Note: We need to add
  more questions later.-->
</quiz>
```

The XML logo consists of the letters "XML" in a bold, white, sans-serif font, centered within a solid orange rectangular background.

[\(Quelle: Wikipedia\)](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd" version="2.0">
  <channel>
    <title>Vorlesung Digitale Medien Wintersemester 2008/09</title>
    <itunes:author>
      Heinrich Hussmann,
      Ludwig-Maximilians-Universität München
    </itunes:author>
    <link>http://www.medien.ifi.lmu.de/lehre/ws0809/dm/</link>
    <itunes:subtitle>
      Eine Einführung in Technologien
      für digitale Medien
    </itunes:subtitle>
    <description>
      Es werden Basiskenntnisse über die physikalischen und
      wahrnehmungsphysiologischen Grundlagen der Realisierung
      digitaler Mediensysteme sowie elementare Techniken der
      digitalen Medienrepräsentation (einschließlich
      Datenkompressionstechniken) vermittelt. Es soll ein
      grundlegendes Verständnis der verschiedenen
      Multimedia-Datenformate und ihrer Vor- und Nachteile
      erworben werden.
    </description>
    <language>de-de</language>
    <copyright>Heinrich Hussmann, LMU</copyright>
    <itunes:category text="Education"></itunes:category>
  </channel>

```

...

(http://www.medien.ifi.lmu.de/lehre/ws0809/dm/dm_podcast.rss)



```
<title>Vorlesung Digitale Medien  
Wintersemester 2008/09</title>
```

```
<itunes:subtitle>  
Eine Einführung in Technologien  
für digitale Medien  
</itunes:subtitle>
```

```
<link>http://www.medien.ifi.lmu.de/lehre/ws0809/dm/</link>
```

Definition des Aufbaus eines XML-Formats in der **DTD (Document Type Definition)**

Eigene Syntax mit zwei Hauptbefehlen:

```
<!ELEMENT Elementname ( Inhaltsbeschreibung )>
```

Elementdefinition (Name, welche anderen Elemente sind enthalten, etc.)

z.B.: **DTD:**
 <!ELEMENT person (vorname, nachname)>
 <!ELEMENT vorname (#PCDATA)>
 <!ELEMENT nachname (#PCDATA)>

XML:
<person>
 <vorname>Hans</vorname>
 <nachname>Müller</nachname>
</person>

Definition des Aufbaus eines XML-Formats in der **DTD (Document Type Definition)**

Eigene Syntax mit zwei Hauptbefehlen:

```
<!ATTLIST Elementname Attributdefinition+ >
```

Attributsdefinition (In welchem Element ist das Attribut enthalten, welche Werte nimmt es an, etc.)

z.B.: DTD:

```
<!ATTLIST person geschlecht (m | w) #REQUIRED>  
<!ATTLIST person alter CDATA #REQUIRED>
```

XML:

```
<person geschlecht="m" alter="48">  
    <vorname>Hans</vorname>  
    <nachname>Müller</nachname>  
</person>
```

Für jedes XML-Format existiert eine entsprechende DTD.

z.B. RSS 2.0 (vereinfachter Ausschnitt):

RSS2-DTD:

<http://www.silmaril.ie/software/rss2.dtd>

```
<!ELEMENT rss (channel)>
<!ATTLIST rss version CDATA #FIXED "2.0">
<!ELEMENT channel (item+)|
  (title,link,description,
  (language|copyright|managingEditor|webMaster|pubDate|lastBuildDate|
  category|generator|docs|cloud|ttl|image|textInput|skipHours|skipDays)*)) >
<!ELEMENT item ((title|description)+,link?,
  (author|category|comments|enclosure|guid|pubDate|source)*)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT category (#PCDATA)>
<!ATTLIST category domain CDATA #IMPLIED>
<!ELEMENT copyright (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT docs (#PCDATA)>
<!ELEMENT image (url,title,link,(width|height|description)*)>
<!ELEMENT language (#PCDATA)>
<!ELEMENT link (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT url (#PCDATA)>
```

```
<channel>
  <title>Vorlesung Digitale Medien Wintersemester 2008/09</title>
  <link>http://www.medien.ifi.lmu.de/lehre/ws0809/dm/</link>
  ...
  <item>
    <title>Informationstheorie, Codierung Teil I</title>
    <description>Es wird eine Einführung ...
```


Für jedes XML-Format existiert eine entsprechende DTD.

z.B. SVG 1.1 (vereinfachter Ausschnitt):

```
<!ENTITY % stdAttrs "id ID #IMPLIED xml:base %URI; #IMPLIED" >
<!ENTITY % Coordinates "CDATA">
...
<!ELEMENT text (#PCDATA|
  desc|title|metadata|tspan|tref|textPath|altGlyph|a|animate|set|
  animateMotion|animateColor|animateTransform %geExt;%textExt;)* >
<!ATTLIST text
  %stdAttrs;
  style %StyleSheet; #IMPLIED
  x %Coordinates; #IMPLIED
  y %Coordinates; #IMPLIED
  dx %Lengths; #IMPLIED
  dy %Lengths; #IMPLIED
  rotate %Numbers; #IMPLIED
  textLength %Length; #IMPLIED
  lengthAdjust (spacing|spacingAndGlyphs) #IMPLIED >
...
<!ELEMENT animate (%descTitleMetadata;%animateExt;) >
<!ATTLIST animate %stdAttrs;
...

```

```
<text x="100" y="100" rotate="45">
  <animate attributeName="x" from="100" to="200" dur="10s"/>
  Ich bin ein Beispieltext!
</text>

```

SVG1.1-DTD:

<http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd>

Element 'A' enthält entweder genau ein Element 'B' oder mind. ein Element 'C':

```
<!ELEMENT A (B|C+)>
```

Element 'A' enthält genau ein Element 'B' und evtl. noch ein Element 'C':

```
<!ELEMENT A (B,C?)>
```

Element 'A' enthält keine Elemente und hat ein optionales Attribut 'B' mit beliebigem Text:

```
<!ELEMENT A EMPTY>
<!ATTLIST A B CDATA #IMPLIED>
```

Element 'A' enthält beliebige Elemente und hat das Attribut 'B' mit den Werten 'c' oder 'd' und das optionale Attribut 'E' mit dem vorgegebenen Wert 'f':

```
<!ELEMENT A ANY>
<!ATTLIST A B (c|d) #REQUIRED
           E CDATA "f">
```

DTD-Übersicht:

<http://www.w3schools.com/dtd/default.asp>

Buch mit Kochrezepten:

- Buch enthält mind. ein Rezept
- Jedes Rezept hat einen Typ, einen Namen, mind. eine Zutat und enthält mind. einen Arbeitsschritt
- Die Arbeitsschritte sind durchnummeriert

```
<kochbuch>
  <rezept>
    <rezepttyp name="Kuchen" />
    <name>Apfelkuchen</name>
    <zutat>Aepfel</zutat>
    <zutat>Mehl</zutat>
    <zutat>Zucker</zutat>
    <schritt nummer="1">
      ...
    </schritt>
    <schritt nummer="2">
      ...
    </schritt>
  </rezept>
</kochbuch>
```

DTD-Übersicht:

<http://www.w3schools.com/dtd/default.asp>

Buch mit Kochrezepten:

- Buch enthält mind. ein Rezept
- Jedes Rezept hat einen Typ, einen Namen, mind. eine Zutat und enthält mind. einen Arbeitsschritt
- Die Arbeitsschritte sind durchnummeriert

```

<kochbuch>
  <rezept>
    <rezepttyp name="Kuchen" />
    <name>Apfelkuchen</name>
    <zutat>Aepfel</zutat>
    <zutat>Mehl</zutat>
    <zutat>Zucker</zutat>
    <schritt nummer="1">
      ...
    </schritt>
    <schritt nummer="2">
      ...
    </schritt>
  </rezept>
</kochbuch>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT kochbuch (rezept+) >
<!ELEMENT rezept
      (rezepttyp, name, zutat+, schritt+) >
<!ELEMENT name (#PCDATA)>
<!ELEMENT zutat (#PCDATA)>
<!ELEMENT schritt (#PCDATA)>
<!ATTLIST schritt nummer CDATA #REQUIRED>
<!ELEMENT rezepttyp EMPTY >
<!ATTLIST rezepttyp name CDATA #REQUIRED>

```

DTD-Übersicht:

<http://www.w3schools.com/dtd/default.asp>

XML-Dateien enthalten nur Inhalt, die Darstellung ist nicht vorgegeben.

Darstellung einer XML-Datei ist möglich über CSS 2.0 mit den Standard-, sowie speziellen neuen Elementen:

```
<leute>
  <person geschlecht="m" alter="48">
    <vorname>Hans</vorname>
    <nachname>M&#x00FC;ller</nachname>
  </person>
  <person geschlecht="w" alter="72">
    <vorname>Anneliese</vorname>
    <nachname>Schmidt</nachname>
  </person>
</leute>
```

```
person
{
    color:#0000FF;
    display: block;
}
person:after
{
    content: "( "
        attr(geschlecht)
        ", "
        attr(alter) " )";
}
```

```
- <leute>
- <person geschlecht="m" alter="48">
  <vorname>Hans</vorname>
  <nachname>Müller</nachname>
</person>
- <person geschlecht="w" alter="72">
  <vorname>Anneliese</vorname>
  <nachname>Schmidt</nachname>
</person>
</leute>
```

Hans Müller (m, 48)
Anneliese Schmidt (w, 72)

Einbindung über `<xml-stylesheet>`-Element:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="present.css"?>

<leute>
  <person geschlecht="m" alter="48">
    <vorname>Hans</vorname>
    <nachname>Müller</nachname>
  </person>
  <person geschlecht="w" alter="72">
    <vorname>Anneliese</vorname>
    <nachname>Schmidt</nachname>
  </person>
</leute>
```

leute.xml

Hans Müller (m, 48)
Anneliese Schmidt (w, 72)

```
person
{
    color:#0000FF;
    display: block;
}
person:after
{
    content: "("
        attr(geschlecht)
        ", "
        attr(alter) ")";
}
```

present.css

Beispieldatei:

wget <http://www.w3schools.com/xml/simple.xml>

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

einfügen

"present.css" erstellen und ändern.

Ergebnisse im Firefox überprüfen.

XML + CSS:

<http://de.selfhtml.org/xml/darstellung/css.htm>

'Generated Content' (CSS 2.0):

<http://www.w3.org/TR/CSS21/generate.html>

Belgian Waffles *only \$5.95!*

two of our famous Belgian Waffles with plenty of real maple syrup

Strawberry Belgian Waffles *only \$7.95!*

light Belgian waffles covered with strawberries and whipped cream

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

<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>two of our famous Belgian Waffles with plenty of real
      maple syrup</description>
    <calories>650</calories>
  </food>
  <food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>light Belgian waffles covered with strawberries and
      whipped cream</description>
    <calories>900</calories>
  </food>
  <food>
    ...
  </breakfast_menu>
```



```
food
{
    display: block;
}

name
{
    color:#FF0000;
    font-weight: bold;
}

price:before
{
    font-style: italic;
    content: "only ";
}
```

```
price:after
{
    content: "!";
}

description
{
    font-style: italic;
    font-size: small;
    display: block;
}

calories
{
    visibility: hidden;
}
```


XML DTD:

Erstellen Sie eine DTD zur Darstellung der Raumbellegung an der Uni. Dabei sollen folgende Aspekte berücksichtigt werden:

- Die Räume werden den verschiedenen Gebäuden zugeordnet.
- Ein Gebäude hat eine Adresse, die aus dem Namen der Straße und einer oder mehrerer Hausnummern besteht. Jedes Gebäude enthält mindestens einen Raum.
- Zu jedem Raum gibt es eine eindeutige Raumnummer (ID) und einen Belegungsplan.
- Optional kann ein Raum auch einen Namen erhalten.
- Ein Belegungsplan enthält Wochentage mit einer Abfolge von Belegungszeiten, angegeben durch Beginn und Dauer. Die Dauer kann auch weggelassen werden, dann wird „60“ (Minuten) als Standardwert angenommen.
- Jeder Belegungszeit sind ein Fach und ein Dozent zugeordnet.

DTD-Übersicht:

<http://www.w3schools.com/dtd/default.asp>

XML/DTD-Validierung:

<http://www.stg.brown.edu/service/xmlvalid/>

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT UniRaumbelegung (Gebaeude+)>
<!ELEMENT Gebaeude (Adresse, Raum+)>
<!ATTLIST Gebaeude Name CDATA #IMPLIED>
<!ELEMENT Adresse (Strasse, Hausnummer+)>
<!ELEMENT Strasse (#PCDATA)>
<!ELEMENT Hausnummer (#PCDATA)>
<!ELEMENT Raum (Belegungsplan)>
<!ATTLIST Raum Nummer CDATA #REQUIRED
           Name CDATA #IMPLIED>
<!ELEMENT Belegungsplan (Wochentag+)>
<!ELEMENT Wochentag (Belegungszeit*)>
<!ATTLIST Wochentag Name CDATA #REQUIRED>
<!ELEMENT Belegungszeit (Fach, Dozent)>
<!ATTLIST Belegungszeit Beginn CDATA #REQUIRED
           Dauer CDATA "60">
<!ELEMENT Fach (#PCDATA)>
<!ELEMENT Dozent (#PCDATA)>
```

DTD-Übersicht:

<http://www.w3schools.com/dtd/default.asp>

XML/DTD-Validierung:

<http://www.stg.brown.edu/service/xmlvalid/>