

# **XML, XSL UND RSS**

VORLESUNG AM SAE INSTITUTE  
DOZENT: BJOERN ZAPADLO

# ABOUT ME

Bjoern Zapadlo  
Konstanz  
34 Jahre

Team Manager / Senior Software Developer  
HolidayCheck AG

Informatik Studium 1999 - 2002  
3 Agenturen in Stuttgart

HolidayCheck International Websites / new Framework  
Neckermann / Thomas Cook

Dozent an der SAE, Dualen Hochschule Stuttgart,  
Hochschule Furtwangen

PHP, Java, Scala, Javascript, CSS, Html, MySQL,  
MongoDB, Elasticsearch, ...

# CONTACT ME

[bjoern.zapadlo@gmail.com](mailto:bjoern.zapadlo@gmail.com)

<http://www.zapadlo.de>

@BjoeZap

[https://www.xing.com/profile/Bjoern\\_Zapadlo](https://www.xing.com/profile/Bjoern_Zapadlo)

<http://de.linkedin.com/pub/bjoern-zapadlo/36/88g/1a5>

Facebook

Google+

# HOLIDAYCHECK AG

Größtes deutsches Meinungsportal für Reise und Urlaub

Vermittlung von Reisen

Sitz in der Schweiz, direkt am Bodensee

Börsennotiert über Tomorrow Focus AG

Existiert seit 1999

Ausgründungen in mehreren europäischen Ländern

Über 300 Mitarbeiter

[HTTP://WWW.HOLIDAYCHECK.DE  
/JOBS](http://www.holidaycheck.de/jobs)

JETZT ABER SCHLUSS MIT DER  
WERBUNG...

# UND IHR?

Name

Erwartungen

Wünsche

# AGENDA

1. Warum?
2. Aufbau
3. Übung XML
4. Validität
5. CSS
6. Suchen
7. Transformation
8. Namensräume
9. XML-Sprachen
10. Einsatzgebiete
11. XHTML
12. RSS
13. Übung RSS
14. Exkurs: JSON
15. Misc / Q&A



# WARUM XML

WER VON EUCH HAT  
HEUTE SCHON XML  
BENUTZT?

AUF FACEBOOK GEWESEN  
ADRESSEN ODER TERMINE AUF  
DEM HANDY SYNCHRONISIERT

GOOGLE EARTH ODER MAPS  
BENUTZT

EIN OFFICE DOKUMENT  
GESCHRIEBEN

IM INTERNET GESURFT

# DAS PROBLEM

## STRUKTURIERUNG UND AUSZEICHNUNG VON TEXT

Das Wort **fett** soll fett sein

Hier möchte ich gerne  
einen Absatz haben

# DIE LÖSUNG?

Das Wort FETTSTARTfettFETTSTOP soll fett sein

Hier möchte ich gerneABSATZeinen Absatz haben

# NAJA, FAST

Eine spezielle Maskierung zur Abgrenzung von Nutz und  
Strukturierungsdaten muss her

# DAS TAG IST <GEBOREN>

# AUSZEICHNUNGSSPRACHEN

Eine Auszeichnungssprache (englisch: Markup Language, ML) dient zur Beschreibung des Inhalts eines Dokumentenformates und teilweise zur Beschreibung des Verfahrens, welches zur Bearbeitung dieser Daten benötigt wird. Ursprünglich dienten die Auszeichnungen im Text als Anweisungen für die Setzer im Drucksatz, mit der Weiterentwicklung in der Typografie für digitale Texte wurden daraus jedoch komplexe Sprachen.

Sind keine Programmiersprachen (PHP, ...) meistens ;)

Können trotzdem sehr komplex sein

Sind Dank des Internets sehr populär

# WAS KANN XML?

## SEMANTIK

Durch die Auszeichnung wird z.B. eine Überschrift zur Überschrift

## MASCHINENLESBARKEIT

Durch die Auszeichnung kann eine (Such)Maschine Texte und deren Bedeutung interpretieren

## MENSCHENLESBARKEIT

Okay, es gibt Sachen, die machen mehr Spass



# WOFÜR

Datenaustausch (in heterogenen Systemen)

Datenbankersatz

# VOR- & NACHTEILE

## VORTEILE

Einfach & Flexibel

Menschen und Maschinenlesbar

Es ist Text

## NACHTEILE

Es ist Text: Encoding, Escaping von Steuerzeichen

Kann Komplex werden

Performance

Overhead

# GESCHICHTE VON XML

Vorgänger SGML (ISO 8879:1986):

Standard Generalized Markup Language

IBM Forschungsprojekt

sehr flexibel aber komplex und performancehungrig

1989 für das CERN HTML

Tim Berners-Lee

Einfach, aber unsauber, extreme Verbreitung

Versionschaos, dann Standardisierung durch das W3C

Parallel XML

Vereinfachtes SGML

XML + HTML = XHTML

# STANDARDS

Standardisiert vom W3C

1.0 (5th Edition)

1.1 (2nd Edition)

# AUFBAU

# DAS ERSTE XML DOKUMENT

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<verzeichnis>
  <titel>Wikipedia Städteverzeichnis</titel>
  <eintrag>
    <stichwort>Genf</stichwort>
    <eintragstext>Genf ist der Sitz von ...</eintragstext>
  </eintrag>
  <eintrag>
    <stichwort>Köln</stichwort>
    <eintragstext>Köln ist eine Stadt, die ...</eintragstext>
  </eintrag>
</verzeichnis>
```

# PHYSISCHES STRUKTUR

## XML-DEKLARATION

Eine XML-Deklaration ist eine Erkennungszeichenfolge im Prolog einer XML-Datei

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

VERSION="1.0"

definiert die Versionsnummer der zugrundeliegenden XML-Spezifikation. Sinnvolle Werte sind momentan 1.0 und 1.1. Der Parameter version muss stets angegeben werden.

ENCODING="ZEICHENKODIERUNG"

bestimmt die Kodierung der XML-Datei. Wird dieser Parameter ausgelassen, wird UTF-8 angenommen. Zu beachten ist, dass der Wert von encoding vom Parser unterstützt werden muss.

STANDALONE="YES|NO"

Eher selten verwendet. Gültige Werte sind yes oder no. Wird der Parameter standalone ausgelassen, wird der Standardwert no angenommen. Der Attributwert yes wird verwendet, wenn ein Dokument über keine DTD verfügt.



# ENTITÄTEN

Die XML-Datei selbst

## ENTITÄTENREFERENZEN

Verknüpfungen auf wiederkehrende Zeichenketten und ganze Dateien

### STANDARD-ENTITÄTEN:

&apos; = '

&amp; = &

&quot; = "

&lt; = <

&gt; = >

# ENTITÄTENREFERENZEN

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!ENTITY c "Chris">
<!ENTITY chap1 SYSTEM "chapter-1.xml">
<verzeichnis>
  <titel>Hallo &c;</titel>
  <!-- Pull in the chapter content: -->
  &chap1;
</verzeichnis>
```

# DTD

Eine Dokumenttypdefinition (englisch Document Type Definition, DTD, auch Schema-Definition oder DOCTYPE) ist ein Satz an Regeln, der benutzt wird, um Dokumente eines bestimmten Typs zu deklarieren.

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE hallo SYSTEM "hallo.dtd">  
<hallo>Hallo Welt!</hallo>
```

# LOGISCHE STRUKTUR

Der logische Aufbau entspricht einer Baumstruktur und ist damit hierarchisch organisiert.

## VERARBEITUNGSANWEISUNG

Verarbeitungsanweisung (engl. processing instruction) oder auch Steueranweisung ist ein Konstrukt von SGML und XML, um Daten abzulegen, die an eine bestimmte (XML-)Anwendung gerichtet ist.

Eine XML-Deklaration ist keine Verarbeitungsanweisung, auch wenn sie genauso aussieht.

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

# ELEMENTE

Auszeichnung via Start- und End-Tag oder einem sog. Empty-Element-Tag

```
<Tag-Name></Tag-Name>  
<Empty-Element />
```

# ATTRIBUTE

Attribute als bei einem Start-Tag oder Empty-Element-Tag geschriebene Schlüsselwort-Werte-Paare (Attribut-Name="Attribut-Wert") für Zusatz-Informationen über Elemente

```
<Tag-Name attribute-key="attribute-value"></Tag-Name>  
<Empty-Element attribute-key="attribute-value" />
```

# TEXT

Normaler Text ;) bzw. die Nutzinformation

```
<Tag-Name>ganz normaler Text</Tag-Name>
```

# CDATA

Mit einem CDATA(Akronym für engl. character data „Zeichendaten“)-Abschnitt wird einem Parser mitgeteilt, dass kein Markup folgt, sondern normaler Text. Dieser wird nicht interpretiert und kann daher auch "illegales" XML (Tags, Sonderzeichen, ...) enthalten.

```
<Tag-Name>  
  <![CDATA[ Inhalt ]]>  
</Tag-Name>  
<Tag-Name>  
  <![CDATA[ Inhalt]] ]]><![CDATA[ >Inhalt ]]>  
</Tag-Name>
```

# KOMMENTARE

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

# REGELN FÜR XML

XML-Deklaration muss vorhanden sein

Baumstruktur muss eingehalten werden:

Wurzelelement darf genau einmal vorhanden sein

Tags müssen geschlossen und die korrekte

Verschachtelung muss eingehalten werden

XML ist „Case Sensitive“ und nicht auf ASCII beschränkt

Kommentare müssen korrekt sein

Der Zeichensatz sollte angegeben werden

## "WOHLGEFORMT"

Wenn keine Regel Verletzt wird.

Prüfung erfolgt durch den XML-Parser



# KLASSIFIZIERUNG

## DOKUMENTZENTRIERT

Das Dokument ist an ein Textdokument angelehnt, das für den menschlichen Leser größtenteils auch ohne die zusätzliche Metainformation verständlich ist. XML-Elemente werden hauptsächlich zur semantischen Markierung von Passagen des Dokuments genutzt, das Dokument ist nur schwach strukturiert. Aufgrund der schwachen Strukturierung ist eine maschinelle Verarbeitung schwierig.

## DATENZENTRIERT

Das Dokument ist hauptsächlich für die maschinelle Verarbeitung bestimmt. Es folgt einem Schema, das Entitäten eines Datenmodells beschreibt und definiert, in welcher Beziehung die Entitäten zueinander stehen,

# SEMISTRUKTURIERT

Semistrukturierte Dokumente stellen eine Art Mischform dar, die stärker strukturiert ist als dokumentzentrierte Dokumente, aber schwächer als datenzentrierte Dokumente.

# ÜBUNG XML

PHP: ZUGRIFF VIA DOM

PHP: ZUGRIFF VIA SIMPLE-XML

JAVASCRIPT: ZUGRIFF VIA AJAX &  
NATIVE JS

JAVASCRIPT: ZUGRIFF VIA JQUERY

# VALIDITÄT

# DEFINITION

Soll XML für den Datenaustausch verwendet werden, ist es von Vorteil, wenn das Format mittels einer Grammatik (z. B. einer Dokumenttypdefinition oder eines XML-Schemas) definiert ist. Der Standard definiert ein XML-Dokument als gültig (oder englisch valid), wenn es **wohlgeformt** ist, den **Verweis auf eine Grammatik** **enthält** und das durch die Grammatik beschriebene **Format einhält**.

# DTD

Eine Dokumenttypdefinition (englisch Document Type Definition, DTD, auch Schema-Definition oder DOCTYPE) ist ein Satz an Regeln, der benutzt wird, um Dokumente eines bestimmten Typs zu deklarieren. Ein Dokumenttyp ist dabei eine Klasse ähnlicher Dokumente, wie beispielsweise Telefonbücher oder Inventurdatensätze.

Die Dokumenttypdefinition besteht dabei aus Elementtypen, Attributen von Elementen, Entitäten und Notationen.

Konkret heißt das, dass in einer DTD die Reihenfolge, die Verschachtelung der Elemente und die Art des Inhalts von Attributen festgelegt wird – kurz gesagt: die Struktur des Dokuments.

Diese können direkt im XML-Dokument definiert werden oder von dort verlinkt werden.

# XML-DOKUMENT

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Person SYSTEM "toll.dtd">

<Person>
  <Name>Manfred Müller</Name>
  <Kind>
    <Name>Olivia Müller</Name>
  </Kind>
  <Kind>
    <Name>Marc Müller</Name>
  </Kind>
</Person>
```

## DTD

```
<!ELEMENT Person (Name, Kind*)>
<!ELEMENT Kind (Name)>
<!ELEMENT Name (#PCDATA)>
```



# AHA...

Diese DTD sagt aus, dass es ein Element Person gibt, mit den Unterelementen Name und Kind, wobei Kind öfter als einmal vorkommen darf.

Desweiteren bestimmt diese DTD, dass Name mit Text oder weiteren Unterknoten gefüllt sein darf während CDATA für reinen Text steht.

# XSD

XML Schema, abgekürzt XSD (XML Schema Definition), ist eine Empfehlung des W3C zum Definieren von Strukturen für XML-Dokumente. Anders als bei den klassischen XML-DTDs wird die Struktur in Form eines XML-Dokuments beschrieben. Darüber hinaus wird eine große Anzahl von Datentypen unterstützt.

XML Schema beschreibt in einer komplexen Schemasprache Datentypen, einzelne XML-Schema-Instanzen (Dokumente) und Gruppen solcher Instanzen. Ein konkretes XML Schema wird auch als eine XSD (XML Schema Definition) bezeichnet und hat als Datei üblicherweise die Endung „.xsd“.

# XML-DOKUMENT

```
...  
<pc-Typ>  
  <name>Ein Name, aber Element ist optional</name>  
  <hersteller>Ein Hersteller, aber Element ist optional</hersteller>  
  <id>123</id>  
</pc-Typ>  
...
```

## XSD

```
<xsd:complexType name="pc-Typ">  
  <xsd:sequence>  
    <xsd:element name="name" type="xsd:string"/>  
    <xsd:element name="hersteller" type="xsd:string"/>  
  </xsd:sequence>  
  <xsd:attribute name="id" type="xsd:integer"/>  
</xsd:complexType>
```

# CSS

# XML UND CSS

Analog zu Html lassen sich auch XML Dokument mit CSS stylen. Die Interpretation erfolgt dann z.B. per Browser, welcher XML darstellen kann. Allerdings ist das ein eher ungebräuchlicher Ansatz.

So sieht es aus...

# XML-DOKUMENT

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  .
  .
  .
</CATALOG>
```

# CSS-DATEI

```
CATALOG {  
    background-color: #ffffff;  
    width: 100%;  
}  
CD {  
    display: block;  
    margin-bottom: 30pt;  
    margin-left: 0;  
}  
TITLE {  
    color: #FF0000;  
    font-size: 20pt;  
}
```

# SUCHEN



# XPATH

Die XML Path Language (XPath) ist eine vom W3-Konsortium entwickelte Abfragesprache, um Teile eines XML-Dokumentes zu adressieren. XPath dient als Grundlage einer Reihe weiterer Standards wie XSLT, XPointer und XQuery.

Ein XPath-Ausdruck adressiert Teile eines XML-Dokuments, das dabei als Baum betrachtet wird.

# XML-DOKUMENT

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<dok>
  <kap title="Nettes Kapitel">
    <pa>Ein Absatz</pa>
    <pa>Noch ein Absatz</pa>
    <pa>Und noch ein Absatz</pa>
    <pa>Nett, oder?</pa>
  </kap>
  <kap title="Zweites Kapitel">
    <pa>Ein Absatz</pa>
    <pa format="bold">Erste Zeile</pa>
    <pa format="bold">Zweite Zeile</pa>
    <pa format="italic">Dritte Zeile</pa>
  </kap>
</dok>
```

# XPATH BEISPIELE

Ausdruck

/dok

/\*

selektiert ...

das erste Element dok

das Wurzel-Element

unabhängig vom Namen

(jedes wohlgeformte XML-

Dokument hat genau ein

Wurzel-Element)

//dok/kap

alle kap-Elemente innerhalb

eines dok Elements

//dok/kap[1]

erstes kap-Element innerhalb

eines dok Elements

//pa

alle pa-Elemente auf allen

Ebenen

//kap[@title='Nettes  
Kapitel']/pa

alle Absätze der Kapitel mit  
Titel „Nettes Kapitel“.

<code>child::*</code>	alle Kindelemente des gegenwärtigen Knotens
<code>child::pa</code>	alle pa-Kinder des gegenwärtigen Knotens
<code>child::text()</code>	alle Textknoten des gegenwärtigen Knotens
<code>.</code>	der gegenwärtige Knoten
<code>/*</code>	alle Unterelemente des gegenwärtigen Knotens
<code>/pa</code>	alle pa-Kinder des gegenwärtigen Knotens
<code>pa</code>	alle pa-Kinder des gegenwärtigen Knotens
<code>attribute::*</code>	alle Attribute des gegenwärtigen Knotens
<code>namespace::*</code>	alle Namespaces des gegenwärtigen Knotens

# TRANSFORMA TION

# UMWANDLUNG VON XML-DOKUMENTEN

# XSLT

XSL Transformation, kurz XSLT, ist eine vollständige Programmiersprache zur Transformation von XML-Dokumenten. Sie ist Teil der Extensible Stylesheet Language (kurz XSL, eine in XML notierte Familie von Transformationssprachen zur Definition von Layouts für XML-Dokumente).

XSLT baut auf der logischen Baumstruktur eines XML-Dokumentes auf und dient zur Definition von Umwandlungsregeln. XSLT-Programme, sogenannte XSLT-Stylesheets, sind dabei selbst nach den Regeln des XML-Standards aufgebaut.

Die Stylesheets werden von spezieller Software, den XSLT-Prozessoren, eingelesen, die mit diesen Anweisungen ein oder mehrere XML-Dokumente in das gewünschte Ausgabeformat umwandeln. XSLT-

XML-DOKUMENT + XSLT-  
STYLESHEET  
=> XSLT-PROZESSOR = NEUES  
DOKUMENT

So sieht es aus...



# XML-DOKUMENT

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  ...
</catalog>
```

# XSLT-stylesheet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Trans
<xsl:template match="/">
  <html><body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32"><th>Title</th><th>Artist</th></tr>
      <xsl:for-each select="catalog/cd">
        <tr><td><xsl:value-of select="title" /></td><td><xsl:value-of selec
        </xsl:for-each>
      </table>
    </body></html>
  </xsl:template>
</xsl:stylesheet>
```

NAMENSRÄU  
ME

# PROBLEMSTELLUNG

Mischung mehrere XML-Sprachen in einem Dokument.

Überschneidung bei den Tag-Names

Wer soll welches Tag bearbeiten

Siehe Telefon-Vorwahl ;)

Vorsicht: DTDs unterstützen dies nicht



# XML- SPRACHEN

Im Zusammenhang mit XML wurden vom W3-Konsortium auf Basis von XML viele Sprachen definiert, welche XML-Ausdrücke für häufig benötigte allgemeine Funktionen anbieten wie etwa die Verknüpfung von XML-Dokumenten. Zahlreiche XML-Sprachen nutzen diese Grundbausteine.

Verknüpfung von XML-Ressourcen: XPointer, XLink und XInclude

Selektion von Daten aus einem XML-Datensatz: XQuery

Datenmanipulation in einem XML-Datensatz: XUpdate

Abfassen von elektronischen Formularen: XForms

Definition von XML-Datenstrukturen: XML Schema (= XSD, XML Schema Definition Language), DTD und RELAX NG



Signatur und Verschlüsselung von XML-Knoten: XML  
Signature und XML-Encryption

Aussagen zum formellen Informationsgehalt: XML Infoset

Formatierte Darstellung von XML-Daten: XSL-FO

Definition zum Methoden- bzw. Funktionsaufruf durch  
verteilte Systeme: XML-RPC

Standardisierte Attribute: XML Base und ID (DTD)

XML-basierte deklarative Programmiersprache: MXML

# EINSATZGEBI ETE

# TEXT

DocBook

DITA

XHTML (XML-konformes HTML)

TEI (Text Encoding Initiative)

NITF (News Industry Text Format)

OPML (Outline Processor Markup Language)

OSIS (Open Scripture Information Standard)

OpenDocument-Austauschformat (OASIS Open  
Document Format for Office Applications)

# GRAFIK

SVG (Vektorgrafiken)

X3D (3D-Modellierungssprache)

Collada (Austauschformat für Daten zwischen  
verschiedenen 3D-Programmen)

# GEODATEN

Geography Markup Language (GML)

GPS Exchange Format (GPX): XML für GPS-Daten

Keyhole Markup Language (KML): Koordinaten-Spezifikation für Google Earth

City Geography Markup Language (CityGML)

OpenStreetMap (OSM)

# MULTIMEDIA

MusicXML (Notendaten, aufgeschriebene Musik)

SMIL (zeitsynchronisierte, multimediale Inhalte)

MPEG-7 (MPEG-7 Metadaten)

Laszlo (LZX)

# SICHERHEIT

Security Assertion Markup Language  
(sicherheitsbezogene Informationen beschreiben und  
übertragen)

XML Signature (XML-Schreibweise für digitale Signaturen)  
XML Encryption

# INGENIEURWISSENSCHAFTEN

AutomationML, ein Format zur Speicherung von  
Anlagenplanungsdaten

CAEX, ein Format zur Speicherung hierarchischer  
Objektinformationen

GSDML, ein Format zur Beschreibung von  
Automatisierungsgeräten, die mit Profinet kommunizieren  
können

IODD, ein Format zur Beschreibung von Sensoren und  
Aktoren

# WEITERE

Webservices (z. B. SOAP, WSDL und WS-\*)

Einbindung von Java-Code in XML-Dokumente (XSP)

Synchronisation von Kalenderdaten SyncML

mathematische Formeln (MathML)

Repräsentation von Graphen (GraphML)

Verfahren im Bereich des Semantischen Webs (RDF,  
OWL, Topic Maps, UOML)

Service Provisioning (SPML)

Austausch von Nachrichten (XMPP)

Finanzberichten wie bspw. Jahresabschlüssen (XBRL)

Automobilindustrie (ODX, MSRSW, AUTOSAR-Templates,  
QDX, JADM)

automatisierter Test z. B. von Schaltkreisen (ATML)

Landwirtschaft (AgroXML)

Verlagswesen (ONIX)

Chemie (CIDX)

XHTML



# DEFINITION

Der W3C-Standard Extensible HyperText Markup Language (erweiterbare HTML; Abkürzung: XHTML) ist eine textbasierte Auszeichnungssprache zur Strukturierung und semantischen Auszeichnung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten. Es ist eine Neuformulierung von HTML 4.01 in XML: Im Gegensatz zu HTML, welche mittels SGML definiert wurde, verwendet XHTML die strengere und einfacher zu parsende SGML-Teilmenge XML als Sprachgrundlage. XHTML-Dokumente genügen also den Syntaxregeln von XML.



# UNTERSCHIEDE ZU HTML

Tags immer klein geschrieben: `<br />`

Start- und Ende-Tag oder Empty-Element-Tag: `<div>`  
`</div>`, `<br />`

Attributwert in Anführungszeichen angeben: `<div`  
`class="bla">`

Attributname als -wert angeben: `<input type="radio"`  
`checked="checked" />`

name Attribut wird durch id ersetzt

Vollständige Kopfdaten:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de">
```

# RSS

RSS ist eine seit dem Anfang des Jahres 2000 kontinuierlich weiterentwickelte Familie von Formaten für die einfache und strukturierte Veröffentlichung von Änderungen auf Websites in einem standardisierten Format.

XML-basiert

Quasi-Standard (zusammen mit Atom)

Feeds sind meist in Channels organisiert

Rich Site Summary vs. RDF Site Summary vs. Really Simple Syndication

# VERSIONEN

1999: RSS 0.90

Erfinder: Netscape, basiert auf RDF (Resource Description Framework)

2000: RSS 0.91 - 0.94

basiert auf einfachem XML

2000: RSS 1.0

wieder RDF-basiert

2002: RSS 2.0

Basis ist 0.91, nicht RDF-basiert, Quasi-Standard, nicht vollständig abwärtskompatibel

# SOFTWARE

Blog-Software, z.B. Wordpress, um einen Feed anzubieten

Firefox für Live-Bookmarks

RSSReader (NewsRob, ...), zum on- und offline lesen

Aggregation mehrere Newsfeeds z.B. per GoogleReader

# AUFBAU (VERSION 2.0)

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
  <channel>
    <title>Titel des Feeds</title>
    <link>URL der Webpräsenz</link>
    <description>Kurze Beschreibung des Feeds</description>
    <language>Sprache des Feeds (z. B. "de-de")</language>
    <copyright>Autor des Feeds</copyright>
    <pubDate>Erstellungsdatum("Tue, 8 Jul 2008 2:43:19")</pubDate>
    <image>
      <url>URL einer einzubindenden Grafik</url>
      <title>Bildtitel</title>
      <link>URL, mit der das Bild verknüpft ist</link>
    </image>

    <item>
      <title>Titel des Eintrags</title>
```

# VERKNÜPFUNG AUS HTML

Man kann eine RSS-Datei in der HTML-Seite, deren Inhalte sie maschinenlesbar enthält, verknüpfen. Dieses Verfahren wurde für RSS nie spezifiziert, allerdings können fast alle Aggregatorprogramme dadurch selbständig die Adresse des RSS-Feeds eines Webangebots herausfinden (genannt auto-discovery). Moderne Browser ermöglichen es dem Seitenbesucher, den so verknüpften RSS-Feed zu abonnieren. Beispielsweise wird in der Adress- oder Statusleiste des Browser-Fensters eine RSS-Schaltfläche angezeigt.

```
<link rel="alternate" type="application/rss+xml"
      title="RSS" href="http://www.example.net/feed.rss" />
```

# ATOM

Atom Syndication Format (ASF) wird entwickelt, um die Nachfolge von RSS antreten

Atom entstand aus dem Bedürfnis heraus, die Vorteile der unterschiedlichen RSS-Formate in einem neuen Format zusammenzufassen und um neue Elemente zu ergänzen. Dabei haben die Entwickler – in überwiegender Mehrzahl Blogger – ASF auch so gestaltet, um den speziellen Bedürfnissen von Weblogs und Nachrichtenseiten gerecht zu werden. Die maßgeblichen Unterstützer von Atom sind in der Industrievereinigung AtomEnabled Alliance organisiert.

Die aktuelle Version des Atom Syndication Formats ist der IETF-Entwurf vom 11. August 2005, welcher von der IESG im August 2005 als Proposed Standard verabschiedet und im Dezember 2005 als RFC 4287 veröffentlicht wurde. Die



# ÜBUNG RSS

PHP: ERSTELLEN EINES RSS 2.0  
FEEDS

PHP: AUSLESEN EINES RSS 2.0  
FEEDS

JAVASCRIPT: AUSLESEN EINES  
RSS 2.0 FEEDS (JQUERY)

# EXKURS: JSON

# Javascript Object Notation

Kurzschreibweise für Javascript Objekte

Sehr einfach

Kaum overhead

Native Unterstützung in vielen Programmiersprachen

# JSON - BEISPIEL

```
{  
  "titel": "Wikipedia Städteverzeichnis",  
  "eintrag": [  
    {  
      "stichwort": "Genf",  
      "eintragstext": "Genf ist der Sitz von ..."  
    },  
    {  
      "stichwort": "Köln",  
      "eintragstext": "Köln ist eine Stadt, die ..."  
    }  
  ]  
}
```

# MISC / Q&A

# FRAGEN?

# THE END

VIELEN DANK FÜR DIE  
AUFMERKSAMKEIT UND  
STIMMUNG!

Made with [reveal.js](#)