



# XML

Grundlagen, mit Praktikum



# Organisatorisches



- Aufteilung in die Praktikumsgruppen
  - Zur Verfügung stehende Termine
  - Ausweichtermine für die Donnerstags-Gruppen
- Leistungsnachweis
  - Per Klausur
  - Erlaubte Hilfsmittel:
    - Skript + handschriftliche (!) Ergänzungen
- Praktikum
  - Mindestens 75% Anwesenheit (hier: 11 von 14)
  - Abgabe der Übungen
    - i.d.R. bis spätestens am Vorabend des nächsten Praktikums
    - Festes Namensschema einhalten, „Abgabe“-Meldungen per email
    - Offensichtliche Kopien werden nicht anerkannt.



- Web-Unterstützung: Homepage des Kurses nutzen!
  - Skripte (PDF)
  - Praktikumsaufgaben
  - Aktuelle Mitteilungen
- eMail nutzen
  - Für Fragen an den Dozenten oder aktuelle Anliegen
  - Option: Verteiler für alle Kursteilnehmer?
- Option Wiki?

Besuch der Seite, sofern  
Netzwerkanschluss vorhanden!



- Das Skript
  - Zielkonflikt Präsentation – Skript zum Nachlesen
  - Ansatz hier:
    - Die meisten Folien sind Präsentationsmaterial, aber:
    - Einige Folien sind dicht beschrieben und zum Nachlesen gedacht
- Hinweise zum Drucken
  - PDFs zu Hause drucken, nicht in der FH (*quotas!*)
  - Mit Änderungen rechnen → möglichst spät ausdrucken
  - Skripte werden inkrementell bereitgestellt



## Übersicht

Vorstellung  
XML - ein Überblick  
(Herkunft, Zweck, Bestandteile)  
Die Spezifikationen



## Vorstellung des Themas, Ziele des Kurses

Eingangsbeispiel  
Methodik  
Vorgeschichte zum Kurs, Ausblick  
Erwartungen der Teilnehmer  
Ziele des Kurses  
Literaturhinweise



```
<Dozent>  
  <Name>  
    <Vorname>Heinz</Vorname>  
    <Nachname>Werntges</Nachname>  
  </Name>  
</Dozent>
```



```
<!-- Vorstellung mittels XML-Text -->
<Dozent>
  <Name>
    <Vorname MI='W'>Heinz</Vorname>
    <Nachname Titel="Dr">
      Werntges
    </Nachname>
  </Name>
  <Beschäftigungsverhältnis Art="Prof"/>
</Dozent>
```



- XML sieht fast aus wie HTML
  - Auszeichnungssprache (*markup language*)
  - gemeinsame Herkunft: SGML
- Allgemeine Unterschiede
  - Klein-/Groß-Schrift unterscheiden!
  - Attribute: Immer in Anführungszeichen ( ' als auch " )
- Elemente
  - anscheinend beliebige Elementtyp-Namen möglich
  - neu: *empty elements*
  - *immer zu schließen!*



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Vorstellung mittels XML-Text -->
<?xml-stylesheet href="demo.css" type="text/css"?>
<!DOCTYPE Dozent SYSTEM "demo.dtd" [
<!ENTITY % ISO-Latin1-chars SYSTEM "iso8859-1.ent">
%ISO-Latin1-chars;
<!ENTITY Abk "Besch&auml;ftigungsverh&auml;ltnis">
]>
<Dozent>
  <Name>
    <Vorname MI='W'>Heinz</Vorname>
    <Nachname Titel="Dr">Werntges</Nachname>
  </Name>
  <&Abk; Art="Prof"/> <!-- *SYNTAXFEHLER* -->
</Dozent>
```



- Induktiv
- Deduktiv
- Anekdotisch
- Wenn möglich - interaktiv!
  
- Unter Einbeziehung  
englischsprachiger Originaldokumente.



- Grundlagen-Teil
  - zu einem Liste V-Kurs im (nächsten) Sommersemester
  - für diverse XML-basierte Technologien
- Thema noch im Aufbau!
  - Erst zum 2. Mal angeboten
  - Mit neuen Komponenten und Akzenten
- Ausblick
  - Folgekurs mit Schwerpunkt „Anwendung“
    - Web Services, APIs, Dokumentenerstellung, e-Commerce





- Vorstellung und Diskussion des Artikels
  - „Take my advice – don't learn XML“  
([http://xml.oreilly.com/news/dontlearn\\_0701.html](http://xml.oreilly.com/news/dontlearn_0701.html))  
von Michael Smith.



- Übersicht
- **XML 1.0** - die Spezifikation
- Optional: CSS2, XHTML
- XML Namespaces, **XML Schema**
- XPath, **XSLT**, XSL-FO
- XPointer, XLink
- Optional: SVG
- Optional: DOM, SAX



- Allgemeine Ziele
  - Solide Grundlagen für weiterführende Arbeiten auf XML-Basis
  - Kenntnis und sicheres Beherrschen der wichtigsten Standards
  - Fähigkeit zu eigener Recherche von Detailfragen in diesen W3C-Dokumenten
  - Entscheidungsgrundlagen für Designfragen



- Publishing-Seite von XML („POP“)
  - Eigene Dokumente auf der Basis vorhandener *document type definitions* (DTDs) bzw. Schemata erstellen und validieren
  - Verständnis von DTDs/Schemata
    - Fähigkeit zur Erweiterung vorhandener DTDs/Schemata
    - Fähigkeit zum Entwurf eigener - einfacher - DTDs.
  - Verarbeitung von XML-Quellen mit Standard-Tools z.B.
    - zu HTML
    - zu PDF
  - „Nagelprobe“:  
Was wird das Quellformat Ihrer Diplomarbeit sein?
    - Noch WORD's DOC-Format - oder besser gleich XML? ...



- Messaging-Seite von XML („MOM“)
  - Grundlagen für Web Services-Technologien erwerben (SOAP, SWA; WSDL, UDDI, ...)
  - Erarbeitung eines realen B2B-Dokumenttyps (z.B. Bestelldatenaustausch)
  - Präzise Typisierung mittels XML Schema



- **Jonathan Pinnock et al.: Beginning XML** 2nd ed.  
Wrox Press Ltd., Birmigham, UK, 2001. ISBN 1-861005-59-8
  - Zum Teil Leitfaden / Quelle von Beispielen in dieser Vorlesung
  - Didaktisch gut, recht vollständig, tiefergehend als der Titel suggeriert.
  - Reihenfolge der Themen offenbar durch Autorenrollen geprägt...
- **Charles Goldfarb, Paul Prescod: Charles F. Goldfarb's XML Handbook** - 4th ed.  
Prentice Hall, Upper Saddle River, NJ, 2002. ISBN 0-13-065198-2
  - Von den SGML- und XML-Vätern selbst
  - Inzwischen auch auf Deutsch erhältlich!
  - Erklärt besonders gut die "warum"-Fragen
  - Große Sammlung von Anwendungsfällen und Ausgangspunkten für weiterführende Themen sowie von Web-Ressourcen.
- **Elizabeth Castro: XML for the World Wide Web.**  
PeachPit Press, Berkeley, 2001. ISBN 0-201-71098-6
  - Kochbuchartig, kompakt, preiswert
  - Fokus auf Web publishing mit XML
- **Michael Kay: XSLT Programmer's Reference** 2nd ed.  
Wrox Press Ltd., Birmigham, UK, 2002. ISBN 1-861005-06-7
  - Die XSLT-"Bibel", vom Autor von `saxon`, des bekanntesten XSLT-Prozessors
- XML Spezifikationen (<http://www.w3c.org>) - mehr dazu später...



# XML - ein Überblick

Herkunft  
Motive der Urheber  
Bestandteile



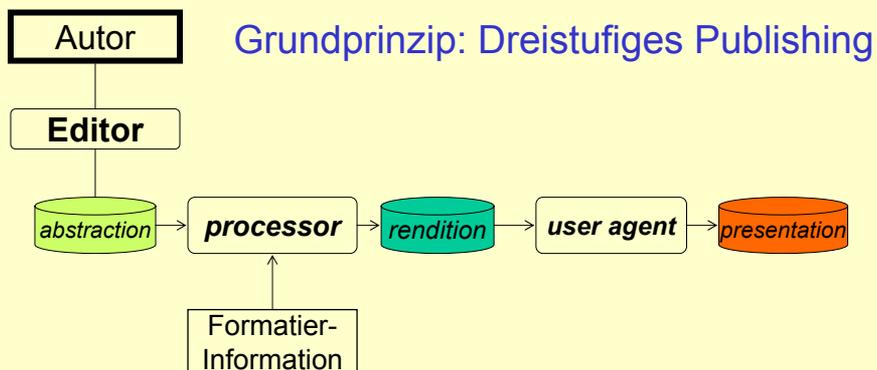
## Herkunft: *Information interchange*



- Austausch zwischen Computersystemen
  - ohne Informationsverlust, d.h. hochstrukturiert
  - von einfachen Zahlen bis zu hochkomplexen Datenstrukturen einerseits ...
  - ... und von menschenlesbaren Dokumenten andererseits, bis hin zu ganzen Buchreihen
  - ... mit der Option zur vollautomatischen Weiterverarbeitbarkeit ...
  - ... wie auch der zur auszugsweisen Übermittlung und Referenzierung bei längeren Dokumenten ...
  - ... sowie der Option zur semantischen, inhaltsbezogenen Suche
- Hier bereits erkennbar: POP & MOM
  - Zwei Enden eines Spektrums der Möglichkeiten

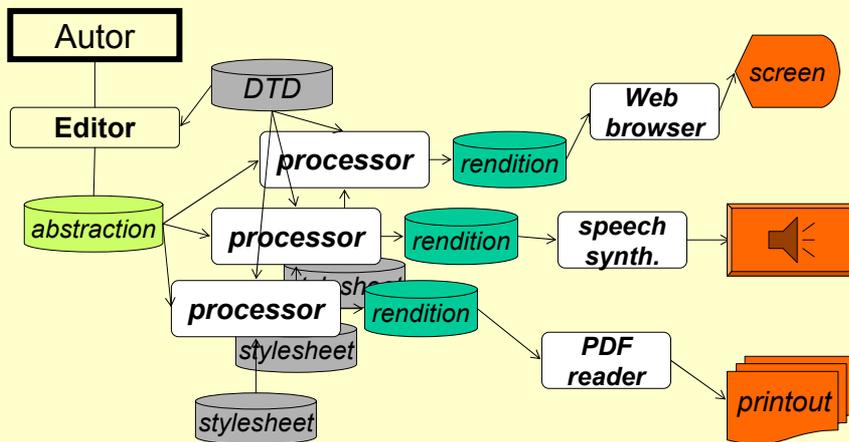


- Die *text processing*-Tradition!
- Der Schlüssel zum Erfolg:  
**Konsequente Trennung der Dokumenterzeugung** in
  - *Abstraction*
    - Die logische Struktur eines Dokuments
    - Beispiele: LaTeX- oder SGML-Quelldaten
  - *Rendition*
    - Die darstellungsorientierte Aufbereitung des Dokuments
    - Beispiele: HTML, troff, RTF, (La)TeX, PS, PDF
  - *Presentation*
    - Das abgelieferte Ergebnis
    - Ausdruck, angezeigte Seite, Tonspur, ...





## Vision: Mehrfach-Verwertung eines abstrakten Dokuments



- Bemerkungen & Warnungen (auch provokative)
  - Die gedankliche Trennung von logischer Ebene und Darstellungsebene erfordert Disziplin und zunächst Mehraufwand, ggf. Verhaltensumstellungen.
  - Contra WYSIWYG - *What you see is **all** you get.*
  - Populäre Publishing/DTP-Werkzeuge und *text processors* trennen gerade nicht bzw. erschweren eine saubere Trennung der logischen von der Darstellungsebene
  - Wer WORD als Schreibmaschinenersatz kennenlernte, wird diese Trennung nur schwer nachvollziehen.
  - Wer beim Verfassen eines Textes schon an's spätere Aussehen denkt, hat die Methode nicht verinnerlicht.
  - Denke „genauso“ über die logische Struktur Deines Dokumentes nach wie über seinen Inhalt - am besten sogar zuerst.



- **EDI, EAI, ATA, B2B, IEC, ...**
  - *EDI: Electronic Data Interchange*
  - *EAI: Enterprise Application Integration*
  - *ATA: Application-to-application*
  - *B2B: Business-to-business ...*
- **IEC: Integrated E-Commerce**
  - Goldfarb's Versuch eines Oberbegriffs
  - Betonung auf vollautomatische Weiterverarbeitung, typisch:
    - Kleine Dokumente, dafür zahlreich, kurzlebig
    - I.d.R. einfache Strukturen (na ja...)
    - Kein / kaum menschlicher Eingriff in die Verarbeitungskette
    - Oft hohe Folgekosten bei Fehldeutungen (produktionskritisch)
    - Daher höchste Präzision notwendig.



- XML: Gemeinsame Grundlage für das breite Spektrum zwischen POP- and MOM-Anwendungen
  - Skeptiker: Noch eine eierlegende Wollmilchsau...
  - Optimisten: Endlich - das „TCP/IP der Datenformate“
- Pole der Diskussion
  - **Überall anwendbar** („Es gibt kaum etwas, das sich nicht mit XML machen läßt“) **versus**
  - **Nirgends wirklich nötig** („Praktisch alle XML-Lösungen wurden schon mit anderen Mitteln gelöst“)
- Versachlichung
  - Mehraufwand jetzt vs. Langfristnutzen ?
  - Entsteht Gewinn durch Vereinheitlichung des Methoden- und Werkzeug-Pools?
  - Ausmaß der Wiederverwertbarkeit semantisch sauber strukturierter Dokumente?



- ... oder: Der Weg zum universellen Datenformat
- 1969: Markup, GML
  - Charles Goldfarb, Ed Mosher, Ray Lorie (IBM)
  - GML = *Generalized Markup Language*
  - Prinzipien:
    - Einheitliche *representation* als *markup*
    - Erweiterbarkeit der *Markup*-Sprache
    - Formale Definition & Beschreibung von Dokumenttypen
- 1974: SGML-Geburtsstunde
  - Erster validierender Parser
- 1986: ISO 8879 (SGML)
  - Ausgereifter, komplexer Industriestandard



- 1989: Tim Berners-Lee
  - gründet HTML auf der Basis von SGML (eigentlich nur GML)
  - aber ohne Erweiterbarkeit und formale Überprüfung
- 199x: Das drohende HTML-Chaos
  - Proprietäre Erweiterungen, inkompatible Browser
  - *W3C-Reaktionen*:
    - *Style sheets* (CSS) - Übernahme eines weiteren GML-Konzepts
    - Erste Ansätze zur standardisierten Erweiterbarkeit von HTML
- 1996: XML Working Group
  - Chair: Jon Bosak, Sun
- 1998-02-10: XML 1.0 - endlich der „große Wurf“?
  - Übernahme auch des dritten Leitgedankens von SGML:
    - Strenge Dokumenttyp-Definitionen und deren Überprüfung
    - Allgemeine Erweiterbarkeit



- 1998-02-10: XML 1.0
  - Autoren (allesamt langjährige Markup-Verfechter):
    - Tim Bray (Netscape),
    - Jean Paoli (Microsoft),
    - C.M. Sperberg-McQueen (TEI / W3C)
  - Übernahme auch des dritten Leitgedankens von SGML:
    - Strenge Dokumenttyp-Definitionen und deren Überprüfung
    - Allgemeine Erweiterbarkeit
  - Endlich der „große Wurf“?
- 2000-10-06: XML 1.0 (SE)
  - Inhaltlich unverändert, nur „*errata*“ berücksichtigt
- 2002-10-15: XML 1.1 (CR)
  - Kleine Erweiterungen insb. im Unicode-Umfeld



- Beibehaltung der wichtigsten Vorzüge von SGML
  - Jedes gültige XML-Dokument ist auch ein gültiges SGML-Dokument
  - 15 Jahre Industriepaxis von SGML werden geerbt
  - Abwärtskompatibilität führt manchmal zu nicht-intuitiven Erweiterungen
- Vereinfachungen für Web-Zwecke
  - minimalistische Tradition
- Weitere Anleihen
  - *Extensible Style Language (XSL)*: abgeleitet von
    - CSS des Web einerseits und
    - ISO's DSSSL (*Document Style Semantics and Specification Language*, sprich „dissel“) andererseits
  - *Extensible Linking Language (XLink)*: abgeleitet von
    - HyTime (ISO Standard zum Verlinken von SGML-Dokumenten)
    - TEI (*Text Encoding Initiative*)-Regeln (akadem. SGML-Umfeld)
  - **Unicode** (<http://www.unicode.org>), ISO 10646
  - **RFC 1766** (*language ID tags*), **ISO 639** (*language name codes*), **ISO 3166** (*country name codes*)



## Die 10 Design-Ziele laut XML 1.0



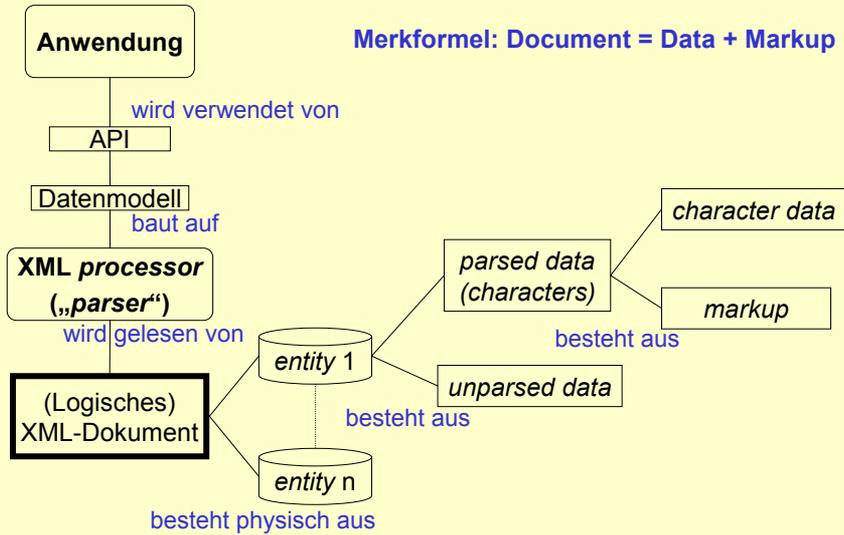
1. *XML shall be straightforwardly usable over the Internet.*
2. *XML shall support a wide variety of applications.*
3. *XML shall be compatible with SGML.*
4. *It shall be easy to write programs which process XML documents.*
5. *The number of optional features in XML is to be kept to the absolute minimum, ideally zero.*
6. *XML documents should be human-legible and reasonably clear.*
7. *The XML design should be prepared quickly.*
8. *The design of XML shall be formal and concise.*
9. *XML documents shall be easy to create.*
10. *Terseness in XML markup is of minimal importance.*



## Die 10 Design-Ziele laut XML 1.0



1. XML soll sich im Internet auf einfache Weise nutzen lassen.
2. XML soll ein breites Spektrum von Anwendungen unterstützen.
3. XML soll zu SGML kompatibel sein.
4. Es soll einfach sein, Programme zu schreiben, die XML-Dokumente verarbeiten.
5. Die Zahl optionaler Merkmale in XML soll minimal sein, idealerweise Null.
6. XML-Dokumente sollten für Menschen lesbar und angemessen verständlich sein.
7. Der XML-Entwurf sollte zügig abgefasst sein.
8. Der Entwurf von XML soll formal und präzise sein.
9. XML-Dokumente sollen leicht zu erstellen sein.
10. Knaptheit von XML-Markup ist von minimaler Bedeutung.



# XML - Die wichtigsten Spezifikationen

Namen, Quellangaben und Kurzbeschreibungen



- **Hinweis zu deutschsprachiger Dokumentation** (zum Nachlesen)
  - Generell sind nur die englischsprachigen W3C-Originaldateien maßgebend. Diese werden auf den folgenden Seiten im Detail benannt.
  - Es gibt ein Projekt, das (nach und nach) diese W3C-Dateien ins Deutsche übersetzt und veröffentlicht. Es besitzt dazu die Genehmigung vom W3C - und ist in dieser Hinsicht die einzige deutschsprachige „offizielle“ Quelle dieser Art.
  - Übersetzte Dateien finden sich unter dem gewohnten URL, wenn man den *domain*-Teil wie folgt systematisch abändert. Beispiel:
    - Original: <http://www.w3.org/TR/REC-xml>
    - Deutsche Version: <http://www.edition-w3c.de/TR/REC-xml>
  - Nur wenige Dokumente sind bereits übersetzt. Was übersetzt ist, ist selten ein sehr aktuelles Dokument. Über die Qualität lassen sich noch keine Aussagen treffen.  
Daher: Besser gleich das englische Original lesen!



- **XML: Extensible Markup Language 1.0**
  - *Autoren:* *Tim Bray, Jean Paoli, C. M. Sperberg-McQueen*
  - *Quelle:* <http://www.w3.org/TR/REC-xml>
  - *Kommentare:*
    - DIE Grundlage schlechthin
    - Präzise, kompakt, vollständig, allgemeingültig.
    - Erfreulich klar geschrieben und gut lesbar - nach einer Eingewöhnung...



- Namespaces in XML

- Autoren: *Tim Bray, Dave Hollander, Andrew Layman*
- Quelle: <http://www.w3.org/TR/REC-xml-names>
- Kommentare:
  - Eine Konvention zur Vermeidung von Namenskollisionen beim Mischen von XML-Dokumentteilen unterschiedlicher Herkunft.
  - Grundlage weiterer Spezifikationen, insb. von
    - XML Schema
    - XPath und XSLT
    - XPointer, XLink



- Associating stylesheets with XML documents

- Autoren: *James Clark*
- Quelle: <http://www.w3.org/TR/xml-stylesheet>
- Kommentare:
  - Dieser - sehr einfache - Standard beschreibt, wie *stylesheets* aus einem XML-Dokument heraus angesprochen (referenziert) werden.
  - Die Methode versteht sich als unabhängig von der benutzten Anwendung wie auch unabhängig von der verwendeten *stylesheet*-Sprache.



- CSS2: Cascading Style Sheets - level 2
  - Autoren: *Håkon Wium Lie, Bert Bos*
  - Quelle: <http://css.nu/pointers/>
  - Kommentare:
    - Dieser *style sheet*-Standard ist bereits über HTML populär geworden und wird bereits von gängigen Browsern unterstützt.
    - Er gestattet auch die „direkte“ Gestaltung von XML-Dokumenten bei deren Anzeige in heutigen Browsern
    - Im XML-Kontext ist CSS noch wichtiger als im HTML-Kontext, da die Browser zwar für HTML verwendbare *default style*-Einstellungen aufweisen, aber XML-Dokumente - wenn überhaupt - nur in einer pauschalen, am hierarchischen Aufbau orientierten Weise darstellen.
    - XML + CSS2 kann in vielen einfachen Fällen bereits befriedigende Ergebnisse auf der Darstellungsseite liefern.
    - Ein viel allgemeingültigerer Ansatz verwendet XSL (XSLT, FO).



- XML Schema Part 0: Primer
  - Autoren: *David C. Fallside*
  - Quelle: <http://www.w3c.org/TR/xmlschema-0>
  - Kommentare:
    - Dies ist *kein* Standard, sondern ein Tutorium.
    - Empfohlene Lektüre vor der Beschäftigung mit den eigentlichen Schema-Spezifikationen.



- XML Schema Part 1: Structures

- Autoren: *David Beech, Murray Maloney, Noah Mendelsohn, Henry S. Thompson*
- Quelle: <http://www.w3.org/TR/xmlschema-1>
- Kommentare:
  - Einer der wichtigsten Standards für anspruchsvolle Entwicklungen auf XML-Basis.
  - Unter Beibehaltung der formalen SGML-Kompatibilität wurde dieser Standard XML beigefügt - und geht dabei weit über SGML hinaus.
  - Ein Schema wird selbst in XML spezifiziert.
  - Schemata ergänzen die traditionellen DTDs um zahlreiche Verallgemeinerungen zur präzisen und erweiterten Definition von Elementtypen.
  - Teile 1 & 2 bilden zusammen die vollständige Schema-Spezifikation.



- XML Schema Part 2: Datatypes

- Autoren: *Paul V. Biron, Ashok Malhotra*
- Quelle: <http://www.w3.org/TR/xmlschema-2>
- Kommentare:
  - Ein reicher Satz vordefinierter Datentypen sowie die Möglichkeit, komplexe eigenen Datentypen zu definieren, bilden einen Schwerpunkt der Schema-Spezifikationen.
  - Die Spezifikation der Datentypen wurde als separater Teil formuliert, denn die Schema-Datentypen lassen sich (per Konvention) auch innerhalb von DTDs verwenden! Allerdings unterstützt nicht jede XML-Software diese erst nachträglich eingeführte Erweiterung.

- XML Information Set (infoset)

- Autoren: *John Cowan, Richard Tobin*
- Quelle: <http://www.w3.org/TR/xml-infoset>
- Kommentare:
  - Dieser Standard legt ein formales Datenmodell für XML-Dokumente fest, d.h. definiert die Datenstrukturen, die ein XML parser erzeugt.
  - Da grundsätzlich über Parser und demnach über derartige Datenstrukturen auf XML-Daten zugegriffen wird - und eben nicht auf den XML-Quelltext - hat dieser Standard eine viel größere Bedeutung als sein bisher geringer Verbreitungsgrad vermuten lässt.
  - Grundlage für XML APIs

- XPath: XML Path Language

- Autoren: *James Clark, Steve DeRose*
- Quelle: <http://www.w3.org/TR/xpath>
- Kommentare:
  - Mit dieser leistungsfähigen und erweiterbaren Sprache können XML-Dokumente durchsucht und gefiltert werden.
  - Sie dient insbesondere zur Adressierung bestimmter Bestandteile von XML-Dokumenten und zur Berechnung von Werten aus diesen.
  - Ein sehr vielseitiger und nützlicher Standard
  - Wird selten isoliert verwendet, sondern bildet die Grundlage weiterer Standards, insb. von XSLT und XPointer sowie XQuery.



- XSLT: XSL Transformations

- *Autoren:* James Clark
- *Quelle:* <http://www.w3.org/TR/xslt>
- *Kommentare:*
  - XSLT ist eine - in XML verfasste - Transformationssprache zur Steuerung der Umwandlung von XML-Dokumenten in
    - (a) andere XML-Dokumenttypen
    - (b) HTML
    - (c) beliebige Textformate
  - Sie entstand durch Abspaltung aus XSL, als man erkannte, dass die Transformation von XML-Dokumenten eine eigenständige und über *stylesheets* hinausreichende Aufgabe ist.
  - XSLT basiert auf XPath



- XSL: eXtensible Stylesheet Language

- *Autoren:* Sharon Adler, Anders Berglund, Jeff Caruso, Stephen Deach, Paul Grosso, Eduardo Gutentag, Alex Milowski, Scott Parnell, Jeremy Richman, Steve Zilles
- *Quelle:* <http://www.w3.org/TR/XSL/>
- *Kommentare:*
  - Diese Spezifikation beschreibt, wie das Formatieren von XML-Dokumenten generell spezifiziert werden soll.
  - Umfangreiches Dokument
  - Erfordert viel Hintergrundwissen zu Seitengestaltung im Allgemeinen!
  - Die Umsetzung dieser Formatierungen bleibt XSLT überlassen.



- XLink: XML Linking Language

- Autoren: *Eve Maler, Steve DeRose*
- Quelle: <http://www.w3.org/TR/xlink>
- Kommentare:
  - XLink legt fest, wie Hyperlinks in XML-Dokumenten definiert werden.
  - XLink geht deutlich über die Möglichkeiten der HTML-Hyperlinks hinaus.



- XPointer: XML Pointer Language

- Autoren: *Ron Daniel Jr., Eve Maler, Steve DeRose*
- Quelle: <http://www.w3.org/TR/xptr>
- Kommentare:
  - Der XPointer-Standard begleitet XLink und beruht auf XPath
  - Mittels XPointer ist es XLink möglich, Hyperlinks bis tief in XML-Dokumente hinein präzise zu definieren.



- XBase: XML Base
  - Autoren: *Jonathan Marsh*
  - Quelle: <http://www.w3.org/TR/xmlbase>
  - Kommentare:
    - Ein einfacher und praktischer Standard zur Veränderung von Basis-URIs in XML-Dokumenten.
    - Eigentlich nur eine Konvention, beruhend auf dem reservierten Attribut `xml:base`.