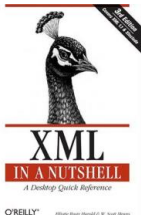


Modul 2

XML Grundlagen

Josef Altmann

<?xml?>



Der vorliegende Foliensatz basiert vorwiegend auf:

Elliotte Rusty Harold, W. Scott Means: XML in a Nutshell: A Desktop Quick Reference, 3rd Edition, O'Reilly, 2005

W3C: XML Technology, www.w3.org/standards/xml

Inhalt

- **XML-Dokument: Aufbau und Bestandteile**
- XML-Prolog: Dokumenteigenschaften
- DTD: Dokumenttypdefinition
- Entities und Verweise

XML-Dokument 1/7

Aufbau

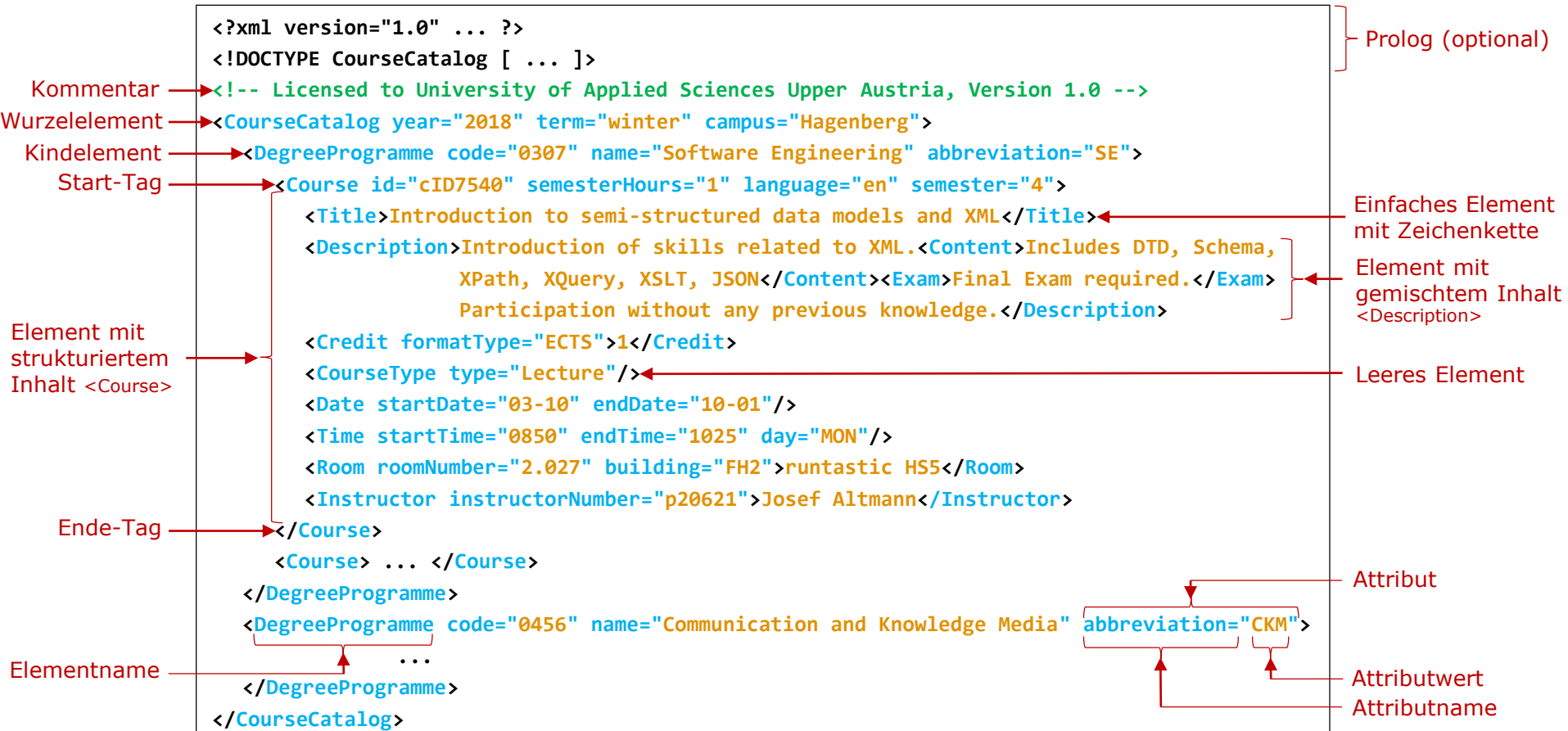
CourseCatalog.xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	XML-Deklaration	Prolog
<!DOCTYPE CourseCatalog [...]>	Dokumenttyp-Deklaration (DTD)	
<pre> <!-- Licensed to University of Applied Sciences Upper Austria, Version 1.0 --> <CourseCatalog year="2018" term="winter" campus="Hagenberg"> <DegreeProgramme code="0307" name="Software Engineering" abbreviation="SE"> <Course id="cID7540" semesterHours="1" language="en" semester="4"> <Title>Introduction to semi-structured data models and XML</Title> <Description>Introduction of skills related to XML.<Content>Includes DTD, Schema, XPath, XQuery, XSLT, JSON</Content><Exam>Final Exam required.</Exam> Participation without any previous knowledge.</Description> <Credit formatType="ECTS">1</Credit> <CourseType type="Lecture"/> <Date startDate="03-10" endDate="10-01"/> <Time startTime="0850" endTime="1025" day="MON"/> <Room roomNumber="2.027" building="FH2">runtastic HS5</Room> <Instructor instructorNumber="p20621">Josef Altmann</Instructor> </Course> <Course> ... </Course> </DegreeProgramme> <DegreeProgramme code="0456" name="Communication and Knowledge Media" abbreviation="CKM"> ... </DegreeProgramme> </CourseCatalog> </pre>		XML-Dokumentinstanz

XML-Dokument 2/7

Beispiel: CourseCatalog.xml

CourseCatalog.xml



XML-Dokument 3/7

Elemente und Attribute: Namensregeln

- Element- und Attributnamen müssen gültige XML-Namen sein:
- `[letter|_|:] [letter|'0..9'|'.'|'-'|'_'|':']*`
- `letter` umfassen A-Z, a-z, sowie andere Schriftzeichen wie bspw. ä, ê, γ, ζ, Ω
- Verwendung von ':' ist Namensräumen vorbehalten
- Zeichenkette "XML" oder "xml" darf nicht am Anfang von Namen auftreten (reserviert für Processing Instructions)
- keine Längenbeschränkung für Namen
- Unterscheidung zw. Klein- und Großschreibung

```
<!-- FALSCH -->  
<Course Catalog>...</Course Catalog>  
<2.Semester>...</2.Semester>  
<Room/Nr>...</Room/Nr>  
<xMlINFO>...</xMlINFO>
```

XML-Dokument 4/7

Elemente und Attribute

- Leere Elemente können in Lang- oder Kurzform geschrieben werden
 - `<Course id="cID8314"></Course>` **oder**
 - `<Course id="cID8314"/>`
- Attributwerte müssen unter Anführungszeichen gesetzt werden
 - `<Course id="cID8314"/>` oder
 - `<Course id='cID8314'/>`
- Element kann beliebige Anzahl von eindeutigen Attributen enthalten
 - `<Course id="cID8314" semesterHours="1" language="en"/>`
 - `<!-- FALSCH -->`
`<Course id="cID8314" semesterHours="1" language="en" language="de"/>`

XML-Dokument 5/7

Kommentare

- Können sich über mehrere Zeilen erstrecken
 - beginnt mit `<!--` und endet mit `-->`
 - zwischen Start-Tag und Ende-Tag eines Elements
 - vor oder nach der Elementwurzel
- Einschränkungen
 - nicht vor dem XML-Prolog und innerhalb von Tags erlaubt
 - keine Schachtelung von Kommentaren erlaubt
 - zwei oder mehr aufeinander folgende "-" im Kommentar nicht erlaubt

```
<!--
```

```
Ein Kommentar kann jede beliebige Art von Text sein, kann  
Leerzeichen, Zeilenumbrüche und <tagname></tagname> enthalten!
```

```
-->
```

XML-Dokument 6/7

Verarbeitungsanweisung (Processing Instruction - PI)

- Verarbeitungsanweisung für eine Anwendung, die ein XML-Dokument verarbeitet
- Der Inhalt einer Verarbeitungsanweisung wird vom XML-Prozessor unverändert an die Anwendung weitergegeben
- Können an beliebiger Stelle im XML-Dokument stehen
- Verarbeitungsanweisung beginnt mit einem Namen
 - identifiziert die Zielanwendung oder gibt der Anweisung einen Namen
 - muss ein gültiger XML-Name sein
- Danach können beliebige Anweisungen stehen

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

Diagramm zur Verarbeitungsanweisung:

- Der Name `xml-stylesheet` ist hervorgehoben.
- Die Attribute `type="text/css"` und `href="format.css"` sind als Verarbeitungsanweisungen gekennzeichnet.

XML-Dokument 7/7

Wohlgeformtes XML

CourseCatalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<CourseCatalog year="2018" term="winter" campus="Hagenberg">
  <DegreeProgramme code="0307" name="Software Engineering" abbreviation="SE">
    <Course id="cID7540" semesterHours="1" language="en" semester="4">
      <Title>Introduction to semi-structured data models and XML</Title>
      <Description>Introduction of skills related to XML.<Content>Includes DTD, Schema,
        XPath, XQuery, XSLT, JSON</Content><Exam>Final Exam required.</Exam>
        Participation without any previous knowledge.</Description>
      <Credit formatType="ECTS">1</Credit>
      <CourseType type="Lecture"/>
      <Date startDate="03-10" endDate="10-01"/>
      <Time startTime="0850" endTime="1025" day="MON"/>
      <Room roomNumber="2.027" building="FH2">runtastic HS5</Room>
      <Instructor instructorNumber="p20621">Josef Altmann</Instructor>
    </Course>
    <Course> ... </Course>
  </DegreeProgramme>
  <DegreeProgramme code="0456" name="Communication and Knowledge Media" abbreviation="CKM">
    ...
  </DegreeProgramme>
</CourseCatalog>
```

... wohlgeformtes XML entspricht syntaktischen Grundregeln von XML!

Inhalt

- XML-Dokument: Aufbau und Bestandteile
- **XML-Prolog: Dokumenteigenschaften**
- DTD: Dokumenttypdefinition
- Entities und Verweise

Prolog 1/4

XML-Deklaration

CourseCatalog.xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	XML-Deklaration
<!DOCTYPE CourseCatalog [...]>	Dokumenttyp-Deklaration (DTD)
<!-- Licensed to University of Applied Sciences Upper Austria, Version 1.0 --> <CourseCatalog year="2018" term="winter" campus="Hagenberg"> <DegreeProgramme code="0307" name="Software Engineering" abbreviation="SE"> <Course id="cID7540" semesterHours="1" language="en" semester="4"> <Title>Introduction to semi-structured data models and XML</Title> <Description>Introduction of skills related to XML.<Content>Includes DTD, Schema, XPath, XQuery, XSLT, JSON</Content><Exam>Final Exam required.</Exam> Participation without any previous knowledge.</Description> <Credit formatType="ECTS">1</Credit> <CourseType type="Lecture"/> <Date startDate="03-10" endDate="10-01"/> <Time startTime="0850" endTime="1025" day="MON"/> <Room roomNumber="2.027" building="FH2">runtastic HS5</Room> <Instructor instructorNumber="p20621">Josef Altmann</Instructor> </Course> <Course> ... </Course> </DegreeProgramme> <DegreeProgramme code="0456" name="Communication and Knowledge Media" abbreviation="CKM"> ... </DegreeProgramme> </CourseCatalog>	

Prolog

XML-Dokumentinstanz

Prolog 2/4

XML-Deklaration ...

- ... ist optional
- ... muss am Anfang der Datei stehen
- ... liefert Informationen an den XML-Parser für die Verarbeitung des Dokuments
- ... legt grundlegende Eigenschaften des XML Dokuments fest:
 - XML-Versionsnummer (**version**)
 - Art der Zeichenkodierung (**encoding**)
 - ggf. Existenz einer DTD (**standalone**)

Prolog 3/4

XML-Deklaration - Parameter

■ **version**

- deklariert verwendete XML-Version
- Parser unterstützen Version "1.0" (ev. auch "1.1")

■ **encoding** (optional)

- definiert die im XML-Dokument verwendete Zeichenkodierung¹
- Standard-Zeichenkodierung ist "UTF-8"

■ **standalone** (optional)

- definiert, ob externe DTD existiert
- Standardwert ist "no"

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
```

Prolog 4/4

XML-Deklaration - Zeichenkodierung

■ XML-Parser

- müssen gemäß XML-Spezifikation intern mit Unicode (UTF-8 oder UTF-16) arbeiten

■ Unicode

- kann alle nationalen Zeichen darstellen
- insgesamt ca. 65.000 Zeichen

■ encoding-Attribut

- Zeichenkodierung der betreffenden XML-Datei
- Fehlt das Attribut, dann wird Kodierung mit Unicode angenommen
- Beachte: XML-Parser müssen gemäß XML-Spezifikation nur Unicode verarbeiten können!

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

```
...  
<Course id="cID7540">  
  <Title>Introduction to XML</Title>  
  ...  
</Course>  
...
```

XML-Dokument

gespeichert als

Unicode
UTF-8

XML-Datei

Inhalt

- XML-Dokument: Aufbau und Bestandteile
- XML-Prolog: Dokumenteigenschaften
- **DTD: Dokumenttypdefinition**
- Entities und Verweise

DTD

Dokumenttypdefinition - DOCTYPE

CourseCatalog.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE CourseCatalog [ ... ]>
<!-- Licensed to University of Applied Sciences Upper Austria, Version 1.0 -->
<CourseCatalog year="2018" term="winter" campus="Hagenberg">
  <DegreeProgramme code="0307" name="Software Engineering" abbreviation="SE">
    <Course id="cID7540" semesterHours="1" language="en" semester="4">
      <Title>Introduction to semi-structured data models and XML</Title>
      <Description>Introduction of skills related to XML.<Content>Includes DTD, Schema,
        XPath, XQuery, XSLT, JSON</Content><Exam>Final Exam required.</Exam>
        Participation without any previous knowledge.</Description>
      <Credit formatType="ECTS">1</Credit>
      <CourseType type="Lecture"/>
      <Date startDate="03-10" endDate="10-01"/>
      <Time startTime="0850" endTime="1025" day="MON"/>
      <Room roomNumber="2.027" building="FH2">runtastic HS5</Room>
      <Instructor instructorNumber="p20621">Josef Altmann</Instructor>
    </Course>
    <Course> ... </Course>
  </DegreeProgramme>
  <DegreeProgramme code="0456" name="Communication and Knowledge Media" abbreviation="CKM">
    ...
  </DegreeProgramme>
</CourseCatalog>
```

XML-Deklaration

Dokumenttyp-Deklaration (DTD)

Prolog

XML-Dokumentinstanz

DTD

Zweck

- Beschreibt Vokabular und Grammatik für eine Menge von XML-Dokumentinstanzen
 - deklariert eine Menge von erlaubten Elementen (Vokabular)
 - definiert ein Inhaltsmodell für jedes Element (Grammatik)
 - deklariert für jedes Element eine Menge von zulässigen Attributen (Attributlisten)

- Stellt Mechanismen zur Verwaltung des Dokumentenmodells bereit
 - Textersetzung
 - Einbindung von Teilen des Dokumentenmodells aus externer Datei

DTD

Charakteristika

- XML-Dokument darf nur eine DTD einbinden
- DTD muss im XML-Dokument nach dem Prolog, jedoch vor der Elementwurzel eingebunden werden
- DTD legt nicht die Elementwurzel eines XML-Dokuments fest
 - dies erfolgt durch das XML-Dokument selbst innerhalb der **DOCTYPE**-Deklaration
 - kann ein beliebiges Element der DTD sein

CourseCatalog.xml

```
<?xml version="1.0" ... ?>
<!DOCTYPE CourseCatalog [ ... <!-- Interne CourseCatalog-DTD --> ... ]>
<CourseCatalog>
  ...
<\CourseCatalog>
```

Definition

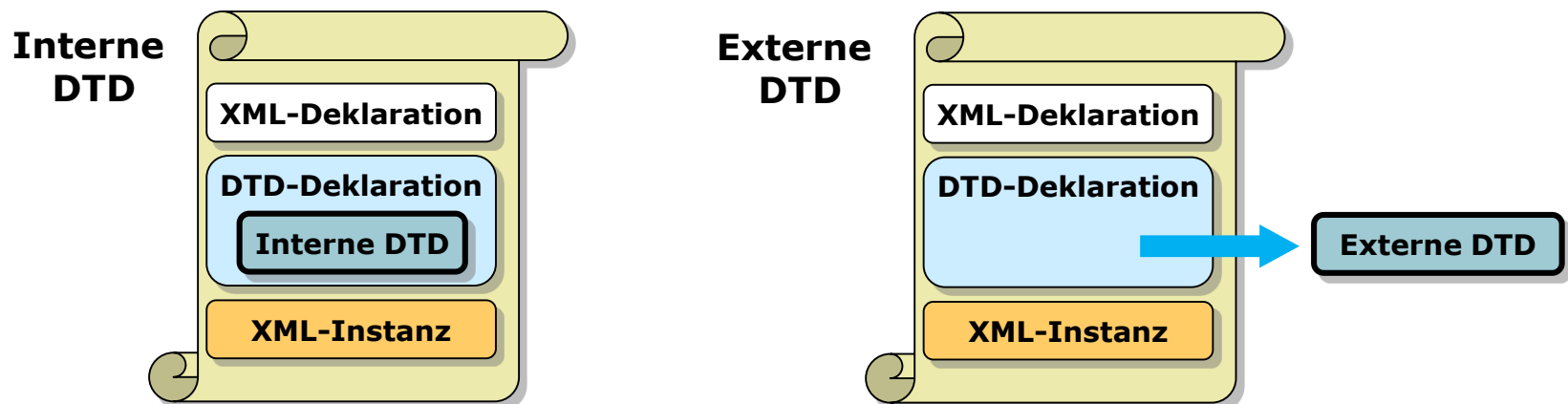
Verwendung

Wurzelement

DTD

Einbindung von DTD's in XML-Dokumenten - 3 Alternativen

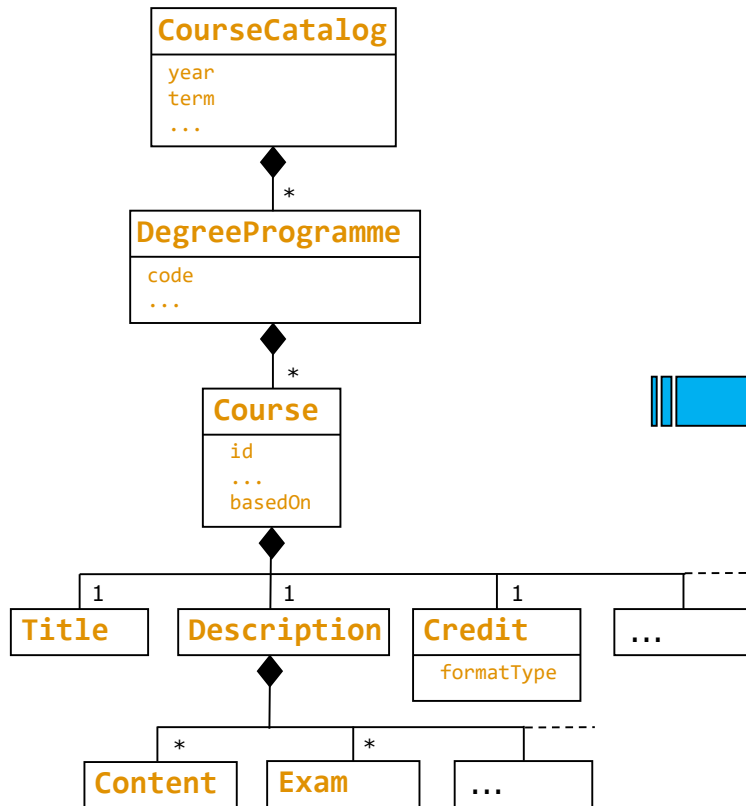
1. Interne DTD, d.h. im Dokument selbst definiert
`<!DOCTYPE CourseCatalog [...]>`
2. Externe DTD, d.h. in eigener Datei (*.dtd)
`<!DOCTYPE CourseCatalog SYSTEM "CourseCatalog.dtd">` oder
`<!DOCTYPE CourseCatalog SYSTEM
"http://www.fh-hagenberg.at/dtds/CourseCatalog.dtd">`
3. Externe & interne DTD, d.h. externe ergänzt interne DTD



DTD

Beispiel: CourseCatalog.dtd

CourseCatalog.uml



Legende:

XML Element	1 : genau eines
XML Attribut	1..* : ein oder mehrere
	* : null oder mehrere
	◆ : besteht aus

CourseCatalog.dtd

```

<!-- CourseCatalog.dtd Version 1.0 -->
<!ELEMENT CourseCatalog (DegreeProgramme*)>
<!ATTLIST CourseCatalog year CDATA #REQUIRED>
<!ATTLIST CourseCatalog term (summer|winter) #REQUIRED>
...
<!ELEMENT DegreeProgramme (Course*) >
<!ATTLIST DegreeProgramme code CDATA #REQUIRED>
...
<!ELEMENT Course (Title, Description, Credit, ...)>
<!ATTLIST Course id ID #REQUIRED>
<!ATTLIST Course basedOn IDREFS #IMPLIED>
...
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Description (#PCDATA|Content|Exam)*>
<!ELEMENT Content (#PCDATA)>
<!ELEMENT Exam (#PCDATA)>
...
<!ELEMENT Credit (#PCDATA)>
<!ATTLIST Credit formatType (ECTS|CP) "ECTS">
...

```

DTD

Inhaltsmodelle von Elementen

■ Inhaltsmodelle:

1. Unstrukturierter Inhalt
2. Strukturierter Inhalt
3. Gemischter Inhalt
4. Leerer Inhalt
5. Beliebiger Inhalt



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

Elementtypdeklaration

Inhaltsmodell legt den erlaubten Inhalt eines Elements fest.

DTD

1. Unstrukturierter Inhalt

DTD:

```
<!ELEMENT Title (#PCDATA)>
```

Dokumentinstanz:

```
<Title>Introduction to semi-structured data models and XML</Title>
```

- Element enthält einfachen Text ohne Kindelemente
- Inhalt auch als *Parsed Character Data* (**PCDATA**) bezeichnet
 - *character data*: einfache Zeichenkette
 - *parsed*: Zeichenkette wird vom Parser analysiert, um End-Tag zu identifizieren
 - reservierte Symbole < und & nicht erlaubt (hierfür Zeichenentitäten < bzw. & benutzen)

DTD

2. Strukturierter Inhalt

DTD:

```
<!ELEMENT CourseCatalog (DegreeProgramme)>
```

Dokumentinstanz:

```
<CourseCatalog>  
  <DegreeProgramme> ... </DegreeProgramme>  
</CourseCatalog>
```

- (Eltern-)Element hat ein oder mehrere Kindelemente
- kein Text vor, nach oder zwischen den Kindelementen
- Elemente können beliebig tief geschachtelt werden

DTD

2. Strukturierter Inhalt - Sequenz

DTD:

```
<!ELEMENT Course (Title, Description)>
```

Dokumentinstanz:

```
<Course>  
  <Title> ... </Title>  
  <!-- Kommentare sind erlaubt -->  
  <Description> ... </Description>  
</Course>
```

- Kindelemente (**Title**, **Description**) sind nur in der angegebenen Reihenfolge erlaubt

DTD

2. Strukturierter Inhalt - Alternative

DTD:

```
<!ELEMENT Course (Title, (Room | Location))>
```

Dokumentinstanz:

```
<Course>  
  <Title> ... </Title>  
  <Room> ... </Room>  
</Course>  
<!-- oder -->  
<Course>  
  <Title> ... </Title>  
  <Location> ... </Location>  
</Course>
```

- Genau eines der alternativen Kindelemente ist erlaubt

DTD

2. Strukturierter Inhalt - Häufigkeitsangaben

- Notwendig (einmal)
`<!ELEMENT CourseCatalog (DegreeProgramme)>`
- Notwendig und wiederholbar (ein- oder mehrmals)
`<!ELEMENT Course (Instructor+)>`
- Optional (null- oder einmal)
`<!ELEMENT Course (Comment?)>`
- Optional und mehrmals (null- oder mehrmals)
`<!ELEMENT DegreeProgramme (Course*)>`
- DTD kennt keine Möglichkeit, Wiederholbarkeit genauer, z. B. "1 bis 5 mal", zu definieren!

DTD

3. Gemischter Inhalt

DTD:

```
<!ELEMENT Description (#PCDATA | Content | Exam | Tool)*>
```

Dokumentinstanz:

```
<Description>  
  Introduction of skills related to XML.<Content>Includes  
  DTD, Schema, XPath, XQuery, XSLT, JSON</Content>  
  <Exam>Final Exam required.</Exam>Participation without  
  any previous knowledge.  
</Description>
```

- alle aufgezählten Elemente können in beliebiger Reihenfolge beliebig häufig vorkommen
- zwischen den Elementen dürfen Zeichenketten vorkommen
- Anmerkung zur DTD-Syntax: **#PCDATA** (am Beginn) und ***** sind obligatorisch für den gemischten Inhalt

DTD

4. Leerer Inhalt

DTD:

```
<!ELEMENT Date EMPTY>
```

Dokumentinstanz:

```
<Date></Date>  
<!-- oder -->  
<Date/>
```

- weder Text noch Kindelement(e) als Inhalt
- Warum leere Elemente?
 - Elemente können Attribute besitzen (wird später behandelt)
 - Bestimmte Stellen in einem Dokument markieren

DTD

5. Beliebiger Inhalt

DTD:

```
<!ELEMENT Date ANY>
```

- Alle in der DTD definierten Elemente können als Inhalt beliebig oft und in beliebiger Reihenfolge vorkommen
- Inhalt muss wohlgeformt sein
- **ANY** sollte nicht verwendet werden (schlechter Stil!), da man damit die Strukturüberprüfung verhindert

DTD

Datentypen für Elementdeklaration

- (nur) drei verschiedene Datentypen:
 - Strukturiertes Inhalt
 - **#PCDATA**: unstrukturierter Inhalt ohne reservierte Symbole < und &.
 - Gemischter Inhalt
 - **EMPTY**: leerer Inhalt, Element kann aber Attribute haben
 - **ANY**: beliebiger Inhalt (strukturiert, unstrukturiert, gemischt oder leer)
- Beachte: Datentypen wie **INTEGER**, **FLOAT** etc. stehen nicht zur Verfügung.

DTD

Attributdeklaration

DTD:

```
<!ATTLIST Elementname  
    Attributname Attributtyp Attributbedingung  
    ...  
>
```

- Attributnamen müssen innerhalb eines Elements eindeutig sein

- Attributbedingung

- Obligatorischer Attributwert
- Optionaler Attributwert
- [Fixer] Vorgabewert

#REQUIRED

#IMPLIED

[#FIXED] "wert"



Kein Vorgabewert
möglich

DTD

Attributdeklaration - Attributtypen

DTD:

```
<!ATTLIST CourseCatalog  
  year CDATA #REQUIRED>
```

Dokumentinstanz:

```
<CourseCatalog year="2018">  
  ...  
</CourseCatalog>
```

- **CDATA** (character data)
 - Beliebige Zeichenkette
 - <, " und & nicht erlaubt (entsprechend zu maskieren)

DTD

Attributdeklaration - Attributtypen

DTD:

```
<!ATTLIST Course
    id          ID          #REQUIRED
    basedOn     IDREFS      #IMPLIED>
```

XML-Dokumentinstanz:

```
<Course id="cID_65"> ... </CourseCatalog>
<Course id="cID_66"> ... </CourseCatalog>
<Course id="cID_75" basedOn="cID_65 cID_66"> ... </CourseCatalog>
```

■ ID, IDREF(S)

- ID gewährleistet Eindeutigkeit von Attributwerten innerhalb eines Dokuments, d.h., zwei Attribute vom Typ ID dürfen niemals gleichen Wert haben
- pro Element ist nur ein Attribut vom Typ ID erlaubt
- IDREF ist eine Referenz auf ein Attribut vom Typ ID
- referentielle Integrität (ungetypt!) wird durch XML-Parser geprüft
- Werte von ID- u. IDREF(S)-Attributen müssen gültige **XML-Namen** sein, d.h. dürfen z.B. nicht mit Zahlen beginnen

DTD

Attributdeklaration - Attributtypen

DTD:

```
<!ATTLIST CourseCatalog  
  year NMTOKEN #REQUIRED>
```

Dokumentinstanz:

```
<CourseCatalog year="2018">  
  ...  
</CourseCatalog>
```

■ NMTOKEN(S)

- XML-Namenstoken sind eine erweiterte Form von XML-Namen
- können zusätzlich mit "0..9", "." und "-" beginnen
- Leerzeichen in XML-Namenstoken sind nicht erlaubt
- NMTOKEN ist ein restriktiverer Datentyp wie CDATA

DTD

Attributdeklaration - Attributtypen

DTD:

```
<!ATTLIST CourseCatalog  
    term (summer|winter) #REQUIRED>
```

Dokumentinstanz:

```
<CourseCatalog term="summer">  
    ...  
</CourseCatalog>
```

■ Aufzählungstyp

- vorgegebene Wertemenge bestehend aus erlaubten Attributwerten (= XML-Namenstoken)

DTD

Dokumenttypdefinition - CourseCatalog.dtd

CourseCatalog.dtd

```

<!-- * Licensed to University of Applied Sciences Upper Austria *
      * http://www.fh-ooe.at/dtds/CourseCatalog.dtd Version 1.0 * -->

<!ELEMENT CourseCatalog (DegreeProgramme*)>
<!ATTLIST CourseCatalog year CDATA #REQUIRED
                        term (summer|winter) #REQUIRED
                        campus (Hagenberg|Linz|Steyr|Wels) #REQUIRED>

<!ELEMENT DegreeProgramme (Course*)>
<!ATTLIST DegreeProgramme code CDATA #REQUIRED
                        name CDATA #REQUIRED
                        abbreviation CDATA #IMPLIED>

<!ELEMENT Course (Title, Description, Credit, CourseType, Date, Time, (Room|Location), Instructor+)>
<!ATTLIST Course id ID #REQUIRED
                semesterHours CDATA #REQUIRED
                language (en|de|fr|es|it) #IMPLIED
                semester CDATA #REQUIRED
                basedOn IDREFS #IMPLIED>

<!ELEMENT Title (#PCDATA)>
<!ELEMENT Description (#PCDATA|Content|Exam|Tool)*>
<!ELEMENT Content (#PCDATA)>
<!ELEMENT Exam (#PCDATA)>
<!ELEMENT Tool (#PCDATA)>

```

DTD

Dokumenttypdefinition - CourseCatalog.dtd

CourseCatalog.dtd

```
<!ELEMENT Credit (#PCDATA)>
<!ATTLIST Credit formatType (ECTS|CP) "ECTS">

<!ELEMENT CourseType EMPTY>
<!ATTLIST CourseType type (Lecture|Seminar|LabSession|PracticeSession|Training) #REQUIRED>

<!ELEMENT Date EMPTY>
<!ATTLIST Date startDate CDATA #REQUIRED
              endDate CDATA #REQUIRED>

<!ELEMENT Time EMPTY>
<!ATTLIST Time startTime CDATA #REQUIRED
              endTime CDATA #REQUIRED
              day (MON|TUE|WED|THU|FRI|SAT) #IMPLIED>

<!ELEMENT Room (#PCDATA)>
<!ATTLIST Room roomNumber CDATA #REQUIRED
              building (FH1|FH2|FH3) #REQUIRED>

<!ELEMENT Location (#PCDATA)>

<!ELEMENT Instructor (#PCDATA)>
<!ATTLIST Instructor instructorNumber ID #REQUIRED>
```

DTD

Gültiges XML-Instanzdokument - CourseCatalog.xml

CourseCatalog.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE CourseCatalog SYSTEM "http://www.fh-hagenberg.at/dtds/CourseCatalog.dtd">
<CourseCatalog year="2018" term="winter" campus="Hagenberg">
  <DegreeProgramme code="0307" name="Software Engineering" abbreviation="SE">
    <Course id="cID7540" semesterHours="1" language="en" semester="4">
      <Title>Introduction to semi-structured data models and XML</Title>
      <Description>Introduction of skills related to XML.<Content>Includes DTD, Schema,
        XPath, XQuery, XSLT, JSON</Content><Exam>Final Exam required.</Exam>
        Participation without any previous knowledge.</Description>
      <Credit formatType="ECTS">1</Credit>
      <CourseType type="Lecture"/>
      <Date startDate="03-10" endDate="10-01"/>
      <Time startTime="0850" endTime="1025" day="MON"/>
      <Room roomNumber="2.027" building="FH2">runtastic HS5</Room>
      <Instructor instructorNumber="p20621">Josef Altmann</Instructor>
    </Course>
    <Course> ... </Course>
  </DegreeProgramme>
  <DegreeProgramme code="0456" name="Communication and Knowledge Media" abbreviation="CKM">
    ...
  </DegreeProgramme>
</CourseCatalog>
```

DTD

Entwurfsentscheidung: Element vs. Attribut

- keine allgemeingültige Antwort, aber Anhaltspunkte
 - Element muss verwendet werden, wenn
 - Inhalt weiter strukturiert werden soll
 - Reihenfolge relevant ist (bei Attributen beliebig!)
 - Elemente mehrmals vorkommen sollen (Attribut kann pro Element nur einmal vorkommen!)
 - Attribut muss verwendet werden, wenn man
 - Aufzählungstyp, Vorgabewert, Fixwert, ID/IDREF einsetzen möchte
- Faustregel
 - Attribute für einfache, unstrukturierte Zusatz- bzw. Metainformationen für Elemente geeignet
 - Alternative Bedingungen sollten durch Attributwerte repräsentiert werden und nicht durch die An- bzw. Abwesenheit von Elementen
 - Elemente sollen für die eigentlichen Daten genutzt werden oder als "künstliches" Gruppierungselement
 - Einheitliche Darstellung mit Elementen eleganter (aber speicherintensiver), Darstellung mit Attributen kompakter

Inhalt

- XML-Dokument: Aufbau und Bestandteile
- XML-Prolog: Dokumenteigenschaften
- DTD: Dokumenttypdefinition
- **Entities und Verweise**

DTD

Zweck

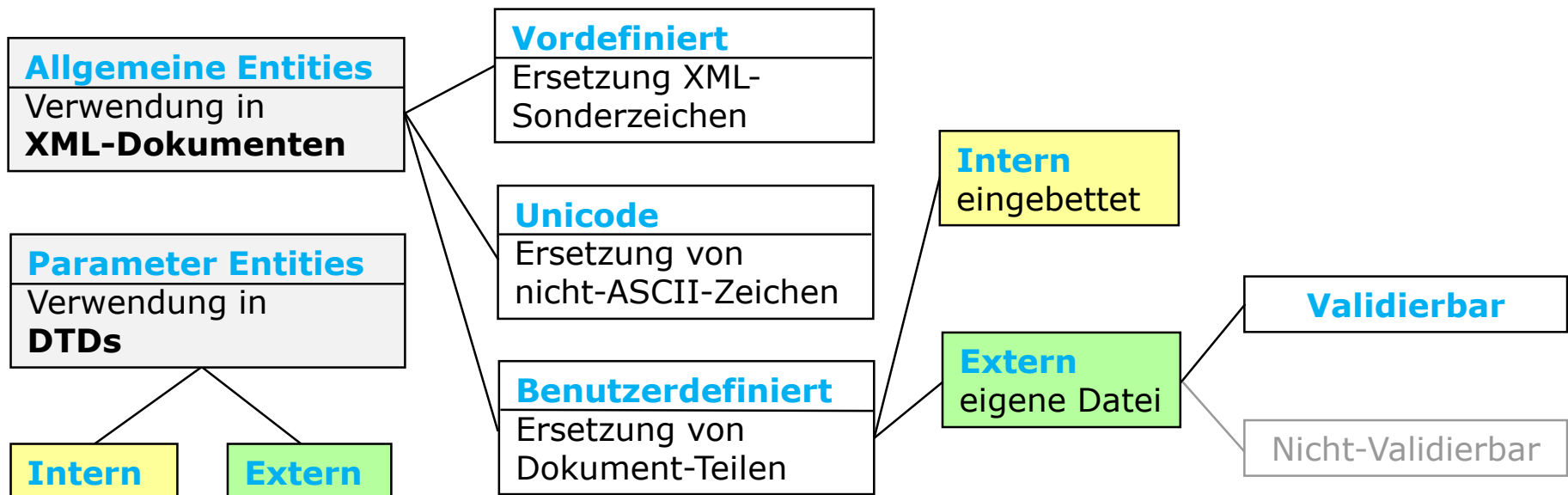
- Beschreibt Vokabular und Grammatik für eine Menge von XML-Dokumentinstanzen
 - deklariert eine Menge von erlaubten Elementen (Vokabular)
 - definiert ein Inhaltsmodell für jedes Element (Grammatik)
 - deklariert für jedes Element eine Menge von zulässigen Attributen (Attributlisten)

- Stellt Mechanismen zur Verwaltung des Dokumentenmodells bereit
 - Textersetzung
 - Einbindung von Teilen des Dokumentenmodells aus externer Datei

Entities

Überblick

- Referenzierbare, mit Namen versehene Textteile
 - eines XML-Instanzdokuments
 - oder einer DTD
- Zweck: Zeichen-/Textersetzung, Modularisierung
- Verarbeitung: Referenzen werden beim Parse-Vorgang expandiert



Entities

Vordefinierte (od. Namendefinierte) Entities

- Zweck:
 - Verwendung von XML-Sonderzeichen (z.B. <, >, &, ', ") im Text
- 5 vordefinierte Entities
 - **&** & (Kaufmännische Und)
 - **<** < (Kleiner als)
 - **>** > (Größer als)
 - **'** ' (Apostrophe: einfache Anführung)
 - **"** " (Doppelte Anführung)

} Verpflichtend innerhalb von #PCDATA
- Beispiel:
 - `<formel>x < y</formel>`
- Verwendung:
 - als Element (#PCDATA) oder Attributwert
- Alternative: **CDATA**-Abschnitt
 - Beispiel:
`<formel>x <![CDATA[<]]> y</formel>`
 - CDATA-Abschnitt darf Begrenzer ']]>' nicht enthalten
 - CDATA-Abschnitt darf nicht geschachtelt werden

Entities

Unicode (od. Zahlendefinierte) Entities

- Zweck:
 - Repräsentation von Zeichen, die am Eingabegerät nicht verfügbar sind
 - Unicode-Standard [<http://www.unicode.org/>]
- Unicode klassifiziert Zeichen nach Buchstaben, Zahlen, Interpunktion, Symbolen (allgemein, technisch, mathematisch) etc.
- Eindeutige Zuordnung Zeichen <-> Zahl
- Unterstützt 25 lebende Sprachen (Cyrillic, Hebrew, Hiragana etc.)
- Insgesamt ca. 60.000 verschiedene Zeichen
- Verwendung:
 - in Element- oder Attributwert
 - beliebige Unicode-Zeichen werden über deren Nummer (dezimal oder hexadezimal) referenziert

```
<?xml version="1.0" encoding="iso-2022-jp"?>
<!DOCTYPE 週報 SYSTEM "weekly-iso-2022-jp.dtd">
<!-- 週報サンプル -->
<週報>
  <業務報告リスト>
    <業務報告>
      <業務名>XML エディターの作成</業務名>
      <業務コード>X3355-23</業務コード>
      <予定項目リスト>
        <予定項目>
          <P>XML エディターの基本仕様の作成</P>
        </予定項目>
      </予定項目リスト>
      <実施事項リスト>
        <実施事項>
          <P>XML エディターの基本仕様の作成</P>
        </実施事項>
        <実施事項>
          <P>競合他社製品の機能調査</P>
        </実施事項>
      </実施事項リスト>
      <問題点対策>
        <P>XML とは何かわからない。</P>
      </問題点対策>
    </業務報告>
  </業務報告リスト>
</週報>
```

© © und ©
stellen das copyright
Zeichen dar!

Entities

Benutzerdefinierte Interne Entities

■ Zweck:

- Platzhalter für wiederkehrende Textfragmente (Text oder wohlgeformter Markup wird mit einem Namen versehen)

■ Deklaration in DTD:

```
<!ENTITY entityName "Ersetzungstext">
```

- Ersetzungstext ist bei der Deklaration anzugeben
- zyklische Referenzen in Entities nicht erlaubt

■ Verwendung im XML-Dokument:

```
&entityName;
```

- in Element- oder Attributwerten des XML-Dokuments

Entities

Benutzerdefinierte Externe Analysierte Entities

■ Zweck:

- Modularisierung des XML-Instanzdokuments

■ Deklaration in DTD:

```
<!ENTITY instructors SYSTEM "InstructorList.xml">
```

- Ersetzungstext befindet sich in eigener Datei

■ Charakteristika:

- prinzipiell wohlgeformt, darf mehrere Elementwurzeln aufweisen
- kann keine DTD einbinden

■ Verwendung:

- in Elementwerten des XML-Dokuments und in Entities selbst
- zyklische Referenzen nicht erlaubt
- NICHT in Attributwerten

Entities

Benutzerdefinierte Externe Nicht-Analyisierte Entities



■ Zweck:

- Einbindung von Dateien, die Nicht-XML-Formate enthalten (z.B. nicht-wohlgeformtes XML, Bilder, Videos)

■ Deklaration in DTD:

```
<!NOTATION JPEG SYSTEM "/usr/local/bin/jpeg_viewer">  
<!ENTITY image SYSTEM "http://www.images.com/image.jpeg" NDATA JPEG>  
<!ELEMENT Picture EMPTY>  
<!ATTLIST Picture src ENTITY #REQUIRED>
```

- **NDATA** kennzeichnet ein nicht-analisiertes (ungeparstes) Entity
- **NOTATION**-Deklaration identifiziert eine Applikation, die Dateien im definierten Format verarbeiten kann

■ Verwendung:

```
<Picture src="image"/>
```

- ausschließlich als Attributwert vom Typ **ENTITY**
- Syntax: Name des Entity in Anführungsstrichen (kein &...;)
- Parser informiert aufrufende Applikation, dass ein nicht-analisiertes Entity existiert - keine Expansion!

■ **(Ausdrucksstärkere) Alternative: W3C's XLink-Standard**

Entities

Benutzerdefinierte, allgemeine Entities - Beispiel

Deklaration

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE CourseCatalog SYSTEM "http://www.fh-hagenberg.at/dtds/CourseCatalog.dtd" [
  <!ENTITY nonBreakingSpace "&#160;">
  <!ENTITY finalExam "Final Exam requiered.">
  <!ENTITY roomRuntastic "<Room roomNumber='2.027' building='FH2'>runtastic HS5</Room>">
  <!ENTITY instructorsCourseID7540 SYSTEM "InstructorList.xml">
  ...
]>
```

Interne Entities

Externes, analysiertes Entity

```
<Instructor instructorNumber="p20621">Josef Altmann</Instructor>
<Instructor instructorNumber="p20622">Norbert Niklas</Instructor>
<Instructor instructorNumber="p20623">Barbara Traxler</Instructor>
```

InstructorList.xml

Verwendung

```
<CourseCatalog year="2018" term="winter" campus="Hagenberg">
  <DegreeProgramme code="0307" name="Software&nonBreakingSpace;Engineering" abbreviation="SE">
    <Course id="cID7540" semesterHours="1" language="en" semester="4">
      <Title>Introduction to semi-structured data models and XML</Title>
      <Description>Introduction of skills related to XML.<Content>Includes DTD, Schema,
        XPath, XQuery, XSLT, JSON</Content><Exam>&finalExam;</Exam>
        Participation without any previous knowledge.</Description>
      <Credit formatType="ECTS">1</Credit>
      <CourseType type="Lecture"/>
      <Date startDate="03-10" endDate="10-01"/>
      <Time startTime="0850" endTime="1025" day="MON"/>
      &roomRuntastic;
      &instructorsCourseID7540;
    </Course>
  </DegreeProgramme>
```


Entities

Parameter Entities

- Zweck:
 - Modularer Aufbau und Wiederverwendung von DTDs
- Deklaration in DTD:

```
<!ENTITY % entityName "Ersetzungstext">
<!ENTITY % entityName SYSTEM "URI">
```

- % mit Leerzeichen

- Anwendung nur in DTD:

```
%entityName;
```

- % ohne Leerzeichen
- Namen und Inhaltsmodellen von Elementen
- Attributdeklarationen

Internes Parameter-Entity

```
<!ENTITY % name
      "(FirstName, LastName)">
...
<!ELEMENT Instructor %name;>
<!ELEMENT Examiner %name;>
```

Externes Parameter-Entity

```
<!ENTITY % courseTypeDefinition
      SYSTEM "CourseTypeDefinition.mod">
...
%courseTypeDefinition;
...
```

Entities

Parameter Entities - Redeklaration (~Überschreiben)

- Ein in einer externen DTD deklariertes Parameter Entity kann in der internen DTD eines XML-Dokuments überschrieben werden
- Damit kann die externe DTD den Erfordernissen einzelner XML-Dokumente direkt innerhalb derselben angepasst werden, ohne die externe DTD ändern zu müssen
- Verwendung Parameter Entity als eine Art "**Customization Hook**"

CourseCatalog.dtd

```
...  
<!ENTITY % name  
    "(FirstName, LastName)">  
...
```

CourseCatalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE CourseCatalog SYSTEM "CourseCatalog.dtd">  
[ <!ENTITY % name  
    "(FirstName, LastName, Middlename)">  
... ]>
```

Entities

Beispiel: CourseCatalog.dtd

Parameter Entities

```
<!-- * Licensed to University of Applied Sciences Upper Austria *
      * http://www.fh-ooe.at/dtds/CourseCatalog.dtd Version 1.0 * -->
<!-- Basic Attributes -->
<!ENTITY % degreeProgrammeAttributes "
    code CDATA #REQUIRED
    name CDATA #REQUIRED
    abbreviation CDATA #IMPLIED">
<!ENTITY % courseAttributes "
    id ID #REQUIRED
    semesterHours CDATA #REQUIRED
    language (en|de|fr|es|it) #IMPLIED
    semester CDATA #REQUIRED
    basedOn IDREFS #IMPLIED">

<!ELEMENT CourseCatalog (DegreeProgramme*)>
...

<!ELEMENT DegreeProgramme (Course*)>
<!ATTLIST DegreeProgramme %degreeProgrammeAttributes;>

<!ELEMENT Course (Title, Description, Credit, CourseType, ...)>
<!ATTLIST Course %courseAttributes;>
...
```

Entities

Beispiel: CourseCatalog.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE CourseCatalog SYSTEM "http://www.fh-hagenberg.at/dtds/CourseCatalog.dtd" > [
  <!ENTITY % courseAttributes "
    id ID #REQUIRED
    semesterHours CDATA #REQUIRED
    language (en|de|fr|es|it) #IMPLIED
    semester CDATA #REQUIRED
    basedOn IDREFS #IMPLIED
    maxNumParticipants CDATA #IMPLIED">
  ]>
  <CourseCatalog year="2018" term="winter" campus="Hagenberg">
    <DegreeProgramme code="0307" name="Software Engineering" abbreviation="SE">
      <Course id="cID7540" semesterHours="1" language="en" semester="4" maxNumParticipants="40">
        <Title>Introduction to semi-structured data models and XML</Title>
        <Description>Introduction of skills related to XML. ...</Description>
        <Credit formatType="ECTS">1</Credit>
        <CourseType type="Lecture"/>
        <Date startDate="03-10" endDate="10-01"/>
        <Time startTime="0850" endTime="1025" day="MON"/>
        <Room roomNumber="2.027" building="FH2">runtastic HS5</Room>
        <Instructor instructorNumber="p20621">Josef Altmann</Instructor>
      </Course>
    </DegreeProgramme>
  </CourseCatalog>

```

Externe DTD

Interne DTD

Redeklaration Parameter Entity

Tipps zum Erstellen von DTDs

- Modularisierung von DTDs (externe DTD)
- Deklaration von logisch zusammengehörigen Elementen in von einander getrennten Abschnitten
- Verwendung von Leerzeichen und Tabulatoren, um DTD zu formatieren und damit lesbarer zu machen
- Verwendung von Kommentaren
- Definition von Parameter-Entities für wiederkehrende Teile von Deklarationen
- Versionierung von DTDs

Zusammenfassung

- DTD unterstützt (u.a. mit dem Entity-Konzept) eine kompakte und relativ schnelle Strukturdefinition von XML-(Text-)Dokumenten
- Schwächen
 - Keine XML-Syntax
 - Fehlende Datentypisierung, insbesondere beim Elementinhalt
 - Unzureichende Kardinalitätsangaben
 - Keine Namensräume
 - Eingeschränkte Wiederverwendbarkeit und Erweiterbarkeit
- XML Schema behebt diese Schwächen