

# XML – Eine Einführung

## XML Dokumente Erstellen

Entities definieren und verwenden

## Entities Definieren Und Verwenden

- Der XML Entity Mechanismus ist
  - Ein zeitsparendes Werkzeug
  - Eine Möglichkeit verschiedene Datentypen in Ihre XML Dokumente einzufügen.
    - Sie können in einem XML Dokument einen häufig verwendeten Block mit XML Text als Entity definieren und diesen Text schnell und einfach überall da einfügen, wo er benötigt wird.
    - Sie können ausserdem eine externe Datei als Entity definieren und somit die Daten aus dieser Datei mit Ihrem Dokument verknüpfen.

## Entities Definieren Und Verwenden

- Sie definieren ein Entity in der Dokumenttyp-Definition (DTD) und verwenden eine Syntax, die derjenigen zur deklaration eines Elements oder Attributs in einem gültigen XML-Dokument gleicht.

## Entities Definieren Und Verwenden

- Lernziele:
  - Sie lernen die grundlegenden Terminologie im Zusammenhang mit Entities und deren Klassifikation kennen.
  - Sie erfahren, wie die verschiedenen Entity Typen deklariert und wie Entities an der richtigen Stelle in ein Dokument eingefügt werden.
  - Sie sehen, wie Zeichenreferenzen und vordefinierte Entities eingesetzt werden können.
  - Schliesslich üben Sie die Verwendung in einem vollständigen XML Dokument.

## Entities Definieren Und Verwenden

### Entity Definitionen Und Klassifikationen

- Der Begriff ‚Entity‘: eine Entity kann sein
  - Das gesamte XML Dokument
  - Eine externe DTD Teilmenge
  - Eine externe Datei, die in der DTD als externe Entity definiert ist und auf die über eine Entity Referenz verwiesen wird
  - Eine in Anführungszeichen eingeschlossene Zeichenkette, die in der DTD als internes Entity definiert ist und auf die über eine Entity Referenz Bezug genommen wird.

## Entities Definieren Und Verwenden

### Entity Definitionen Und Klassifikationen

- Der Begriff ‚Entity‘ im engeren Sinne:
  - Eine Entity ist eine externe Datei  
oder
  - eine in Anführungszeichen eingeschlossene  
Zeichenkette,
    - die in der DTD des Dokuments als Entities definiert sind und  
auf die im Dokument über Entity Referenzen verwiesen wird.

# Entities Definieren Und Verwenden

## Entity Definitionen Und Klassifikationen

### Beispiel:

```
<!DOCTYPE ARTICLE
[
  <!ELEMENT ARTICLE (TITLEPAGE, INTRODUCTION, SECTION*)>
  <!ELEMENT TITLEPAGE (#PCDATA)>
  <!ELEMENT INTRODUCTION (#PCDATA)>
  <!ELEMENT SECTION (#PCDATA)>

  <!ENTITY topics SYSTEM "Topics.xml">
  <!ENTITY title "A Short History of XML">
]
>
```

# Entities Definieren Und Verwenden

## Entity Definitionen Und Klassifikationen

### Beispiel:

```
<INTRODUCTION>
```

Dieser Artikel gefasst sich mit folgenden Themen:

&topics;

```
</INTRODUCTION>
```

### Beispiel:

```
<TITLEPAGE>
```

Titel: &title;

Autor: Michel von Löhneberg

```
</TITLEPAGE>
```

Der Mechanismus ist ähnlich dem #define in C/C++.



Entities Definieren Und Verwenden  
Entity Definitionen Und Klassifikationen

Das vollständige Beispiel finden Sie auf dem Server /  
der CD

- Artikel.xml
- ArtikelSystem.dtd (externe DTD Beschreibung)
- Topics.xml

in der Datei Artikel.dtd enthält die vollständige DTD  
Beschreibung, so wie sie in einer XML Datei (internes  
DTD) stehen würde.

Sie können das Ergebnis mit dem Internet Explorer  
ansehen.

XML Spy zeigt Ihnen, dass die Syntax korrekt ist.

Falls Sie die Elemente <INTRODUCTION> und  
<SECTION> weglassen wird das Dokument immer noch

## Entities Definieren Und Verwenden

### Entity Definitionen Und Klassifikationen - Entity Typen

- Es gibt drei Möglichkeiten Entities zu definieren:
  - Allgemeine und Parameter Entities
  - Interne und Externe Entities
  - Geparste und nicht geparste Entities

## Entities Definieren Und Verwenden

### Entity Definitionen Und Klassifikationen - Entity Typen

- Es gibt drei Möglichkeiten Entities zu definieren:
  - Allgemeine und Parameter Entities
    - Ein *allgemeines Entity* enthält Dokumentinhalt, also XML Text, sonstigen Text oder Daten, die Sie innerhalb des Dokumentelements verwenden können.  
Bei den Beispielen vorne (*topics* und *title*) handelt es sich um allgemeine Entities
    - Ein *Parameter-Entity* enthält XML-Text, der innerhalb der DTD eingefügt werden kann.

In der XML Spezifikation beschreibt der nicht qualifizierte Begriff *Entity* ein allgemeines Entity.

## Entities Definieren Und Verwenden

### Entity Definitionen Und Klassifikationen - Entity Typen

- Es gibt drei Möglichkeiten Entities zu definieren (2):
  - Interne und Externe Entities
    - Ein *internes Entity* ist in einer in Anführungszeichen eingeschlossenen Zeichenkette enthalten (wie die / das Entity *title* im vorigen Beispiel.  
Ein *externes Entity* ist in einer separaten Datei enthalten (wie das Entity *topics* im vorherigen Abschnitt / Beispiel)

## Entities Definieren Und Verwenden

### Entity Definitionen Und Klassifikationen - Entity Typen

- Es gibt drei Möglichkeiten Entities zu definieren(3):
  - Geparste und nicht geparste Entities

- Ein *geparstes Entity* besteht aus XML Text (Zeichendaten, Markup oder beidem).

Fügen Sie eine Referenz auf ein geparstes Entity in ein Dokument ein, dann wird die Referenz durch den Inhalt des Entity ersetzt (auch *Ersetzungstext* genannt), der zu einem Bestandteil des Dokuments wird.

Der XML Parser parst den Inhalt des Entity genauso wie Text, den Sie direkt in das Dokument eingetippt haben.

Beide Beispiele vorne (*title* und *topics*) sind geparste Entities.

## Entities Definieren Und Verwenden

### Entity Definitionen Und Klassifikationen - Entity Typen

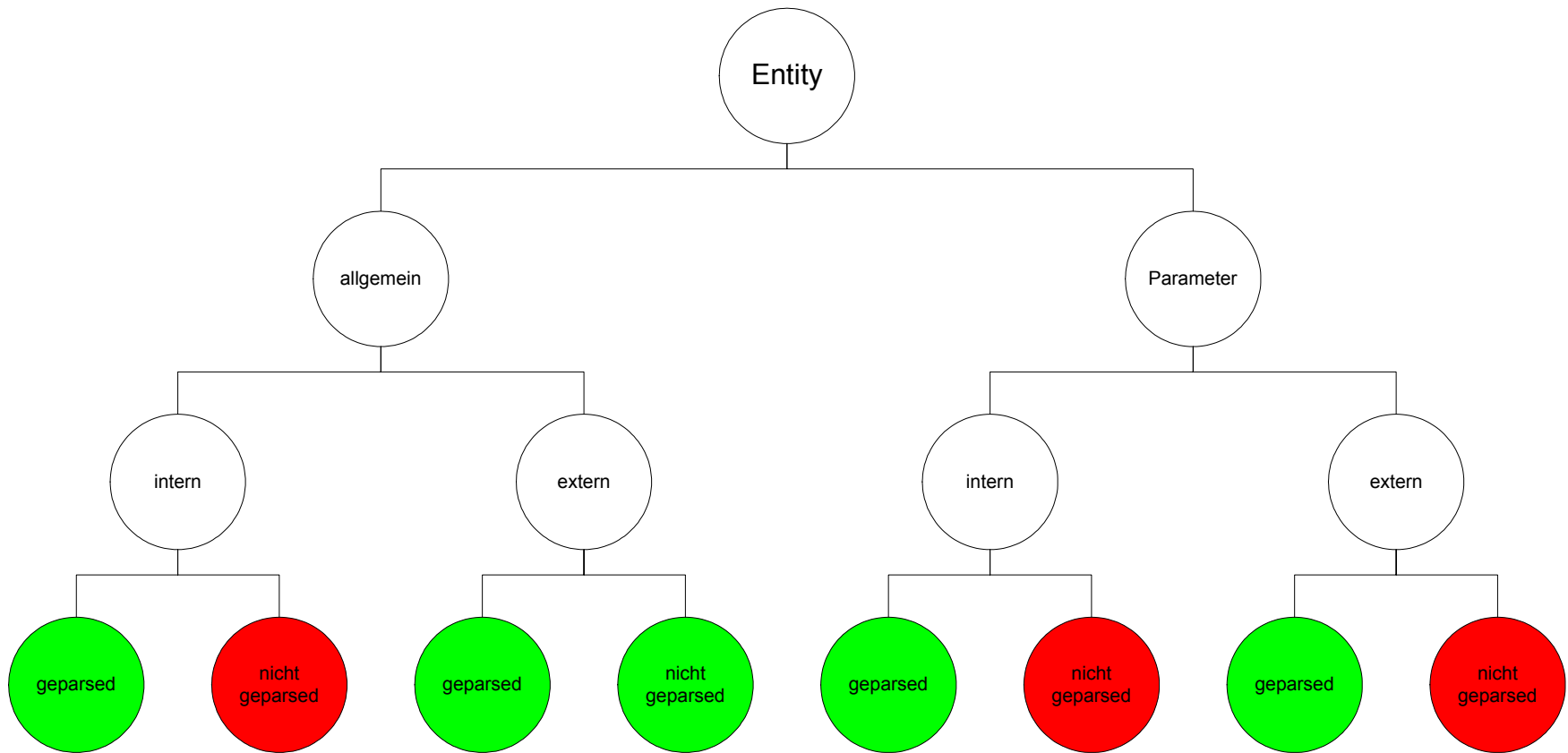
- Es gibt drei Möglichkeiten Entities zu definieren(4):
  - Geparste und nicht geparste Entities
    - Ein *nicht geparstes Entity* kann jeden Datentyp enthalten: XML-Daten oder häufiger XML-fremde Daten. XML-fremde Daten bestehen entweder aus Textdaten (wie etwa einem Titel) oder aus sonstigen Daten (wie etwa Grafikdaten für ein Bild).

Da ein nicht geparstes Entity gewöhnlich kein XML enthält, werden dessen Inhalte nicht direkt über eine Entity Referenz in das Dokument eingefügt.

Allerdings können Sie den Namen des Entity einem Attribut vom Typ ENTITY oder ENTITIES zuweisen, damit die Anwendung auf den Namen und die Beschreibung des Entity zugreifen und die Daten verarbeiten kann.

# Entities Definieren Und Verwenden

## Entity Definitionen Und Klassifikationen - Entity Typen



## Entities Definieren Und Verwenden

### Entity Definitionen Und Klassifikationen - Entity Typen

- XML stellt die drei rot gekennzeichneten Entities nicht zur Verfügung. Daher besitzt XML tatsächlich nur fünf Entity Typen, deren Definition und Verwendung Sie jetzt kennen lernen:
  - Allgemein, intern geparsed
  - Allgemein, extern geparsed
  - Allgemein, extern nicht geparsed
  - Parameter, intern geparsed
  - Parameter, extern geparsed



## Entities Definieren Und Verwenden

### Allgemeine Entities Deklarieren

- Sie erstellen ein Entity, indem Sie es in der DTD des Dokuments deklarieren
  - Dazu verwenden Sie einen Typ der Markup Deklaration, der derjenigen zur Deklaration von Elementen und Attributen gleicht.

## Entities Definieren Und Verwenden

### Ein Allgemeines Internes Geparstes Entity Deklarieren

- Eine Deklaration eines allgemeinen internen geparsten Entity besitzt das folgende Format
  - `<!ENTITY EntityName EntityWert>`
  - *EntityName* bezeichnet den Namen des Entity. Sie können einen beliebigen Namen wählen, sofern er diesen Regeln genügt:
    - Der Name muss mit einem Buchstaben oder einem Unterstrich ( `_` ) beginnen, dem null oder mehrere Buchstaben, Zahlen, Punkte, Bindestriche oder Unterstriche folgend können.
    - Das Entity darf den gleichen Namen wie ein Parameter Entity, ein Attribut oder ein Element im Dokument tragen (allg. & Parameter Entities. Gehören zu unterschiedlichen Namensräumen)
    - Gross- und Kleinschreibung muss beachtet werden

## Entities Definieren Und Verwenden

### Ein Allgemeines Internes Geparstes Entity Deklarieren

- Eine Deklaration eines allgemeinen internen gearsten Entity besitzt das folgende Format
  - `<!ENTITY EntityName EntityWert>`
  - *EntityWert* ist der Wert eines Entity:
    - Die Zeichenkette kann in Apostrophe (‘) oder in Anführungszeichen („) eingeschlossen sein.
    - Die Zeichenkette darf das Zeichen nicht enthalten, in das sie eingeschlossen ist.
    - In der Zeichenkette darf das Et Zeichen (&) nur zu Beginn einer Zeichen- oder allgemein Entity Referenz enthalten sein. Auch das Prozentzeichen (%) ist nicht erlaubt.
    - Der Inhalt der Zeichenkette muss innerhalb der Umgebung, in der er eingefügt wird, gültig sein.

## Entities Definieren Und Verwenden

### Ein Allgemeines Internes Gearstes Entity Deklarieren

- Beispiel (selbes Beispiel wie weiter vorne):

```
<!DOCTYPE ARTICLE
[
  <!ELEMENT ARTICLE (TITLEPAGE, INTRODUCTION,
SECTION*)>
  <!ELEMENT TITLEPAGE (#PCDATA)>
  <!ELEMENT SUBTITLE (#PCDATA)>
  <!ELEMENT INTRODUCTION (#PCDATA)>
  <!ELEMENT SECTION (#PCDATA)>

  <!ENTITY topics SYSTEM "Topics.xml">
  <!ENTITY title "A Short History of XML
    <SUBTITLE>The Future Language of the Internet
    </SUBTITLE>" >
]
>
```

## Entities Definieren Und Verwenden

### Ein Allgemeines Internes Geparstes Entity Deklarieren

- **Beispiel:**

```
<!ENTITY title "A Short History of XML  
  <SUBTITLE>The Future Language of the Internet  
  </SUBTITLE>" >
```

- Das Entity *title* enthält Zeichendaten sowie ein Element (`<SUBTITLE>`). Nach den Deklarationen in der DTD kann dieser Inhalt nur innerhalb eines `TITLEPAGE` Elements legal eingefügt werden:
- `<TITLEPAGE>`  
 Titel; `&title;`  
 Autor: Michael Young  
`</TITLEPAGE>`

Der XML Prozessor ersetzt die Entity Referenz (`&title`) durch den Inhalt des Entity und verarbeitet die Inhalte genau wie Daten.

## Entities Definieren Und Verwenden

### Ein Allgemeines Externes Geparstes Entity Deklarieren

- Die Deklaration eines allgemeinen externen geparsten Entity besitzt das folgende Format:
  - `<!ENTITY EntityName SYSTEM SystemLiteral>`
  - *EntityName* :bezeichnet den Namen des Entity.
    - Sie können einen beliebigen Namen wählen, sofern er den allgemeinen Benennungsregeln für Entities genügt.
  - *SystemLiteral* : ist ein Systemliteral, das den Ort der Datei mit den Entity Daten beschreibt. Das Systemliteral ist entweder in Apostrophe oder Anführungszeichen eingeschlossen und kann jedes Zeichen enthalten, abgesehen von dem, in das es eingeschlossen ist.
    - Das Systemliteral gibt den Uniform Resource Identifier (URI) der Datei an, welche die Entity Daten enthält.  
`<!ENTITY beispiel SYSTEM „http://example.com/beispiel.xml“>`

## Entities Definieren Und Verwenden

### Ein Allgemeines Externes Nicht Geparstes Entity Deklarieren

- Die Deklaration eines allgemeinen externen geparsten Entity besitzt das folgende Format:
  - `<!ENTITY EntityName SYSTEM SystemLiteral NDATA Name>`
  - *EntityName* :bezeichnet den Namen des Entity.
    - Sie können einen beliebigen Namen wählen, sofern er den allgemeinen Benennungsregeln für Entities genügt.
  - *SystemLiteral* : ist ein Systemliteral, das den Ort der Datei mit den Entity Daten beschreibt. Das Systemliteral ist entweder in Apostrophe oder Anführungszeichen eingeschlossen und kann jedes Zeichen enthalten, abgesehen von dem, in das es eingeschlossen ist.
  - NDATA zeigt an, dass die gesamte Datei nicht geparste Daten enthält.
  - *Name* bezeichnet den Namen der Notation, die in der DTD deklariert ist

## Entities Definieren Und Verwenden

### Ein Allgemeines Externes Nicht Geparstes Entity Deklarieren

– Beispiel:

- Die DTD im folgenden Beispiel definiert die Datei herrDerRinge.gif als externe nicht geparste Entity namens *ringe*
- ```
<?xml version=„1.0“?>
<!DOCTYPE BOOK
[
  <!ELEMENT BOOK (TITLE, AUTHOR, COVERIMAGE)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT AUTHOR (#PCDATA)>
  <!ELEMENT COVERIMAGE EMPTY>
  <!ATTLIST COVERIMAGE Source ENTITY #REQUIRED>

  <!NOTATION GIF SYSTEM „ShowGif.exe“>
  <!ENTITY ringe SYSTEM „herrDerRinge.gif“ NDATA GIF>
]
>
```



## Entities Definieren Und Verwenden

### Ein Allgemeines Externes Nicht Geparstes Entity Deklarieren

– Beispiel:

- ```
<?xml version=„1.0“?>
<!DOCTYPE BOOK
...
>
<BOOK>
  <TITLE>Der Herr der Ringe</TITLE>
  <AUTHOR>John Ronald Reuel Tolkien</AUTHOR>
  <COVERIMAGE Source=„ringe“/>
</BOOK>
```

- Der Parser analysiert diese Daten nicht weiter, er stellt die Daten (den Link) einfach zur Verfügung.

## Entities Definieren Und Verwenden

### Ein Allgemeines Externes Nicht Gearartes Entity Deklarieren - Notation

- Eine **Notation** beschreibt ein bestimmtes Datenformat. Sie besitzt die folgende allgemeine Form:
  - `<!NOTATION NotationsName SYSTEM SystemLiteral>`
  - *NotationsName*:
    - Bezeichnet den Notationsnamen (mit den üblichen Namensregeln)
  - *SystemLiteral*:
    - Entweder in ‘ oder „ eingeschlossener externer Name (Literal)
    - Eine URI eines Programms:  
`<!NOTATION BMP SYSTEM „Pbrush.exe“>`  
`<!NOTATION GIF SYSTEM „http://grafik.com/ShowGif.exe“>`
    - Eine URI eines Dokuments:  
`<!NOTATION MEIN_FORMAT SYSTEM „http://test.com/a.html“>`
    - Eine einfache Beschreibung des Formats:  
`<!NOTATION GIF SYSTEM „Graphics Interchange Format“>`

## Entities Definieren Und Verwenden

### Parameter-Entities Deklarieren

- Sie deklarieren ein Parameter-Entity mit einem Typ der Markup-Deklaration, der derjenigen zur Deklaration von allgemeinen Entities gleicht
  - Im folgenden besprechen wir, wie
    - Ein internes geparstes Parameter Entity deklariert wird
    - Ein extern geparstes Parameter Entity deklariert wird

## Entities Definieren Und Verwenden

### Ein Internes Gepaartes Parameter Entity Deklarieren

- Format:
  - `<!ENTITY & EntityName EntityWert>`
  - *EntityName* – Regeln:
    - Der Name muss den üblichen Namensregeln gehorchen.
    - Das Entity darf den gleichen Namen wie ein allgemeines Entity im Dokument tragen (weil sie unterschiedlichen Namensräumen angehören). Das Entity darf auch den gleichen Namen wie ein Element oder ein Attribut haben.
    - Gross- und Kleinschreibung müssen beachtet werden
  - *EntityWert* – Regeln:
    - Die Zeichenkette kann in Apostrophe oder Anführungszeichen eingeschlossen sein.
    - Die Zeichenkette darf das Zeichen nicht enthalten, in das sie eingeschlossen ist.
    - & ist nur zu Beginn, % ist nicht erlaubt.

# Entities Definieren Und Verwenden

## Ein **Internes** Geparstes Parameter Entity Deklarieren

- Beispiel

```
<!DOCTYPE BOOK
[
  <!ENTITY % author
    "<!-- Autoreninformationen -->
    <!ELEMENT AUTHOR (#PCDATA)>
    <!ATTLIST AUTHOR Nationality CDATA 'Amerikaner'>"
  >

  <!ELEMENT BOOK (TITLE, AUTHOR)>
  <!ELEMENT TITLE (#PCDATA)>
  %author;
]
>
```

## Entities Definieren Und Verwenden

### Ein **Internes** Geparstes Parameter Entity Deklarieren

- Diese DTD ist äquivalent zu

```
<!DOCTYPE BOOK
[
  <!ELEMENT BOOK (TITLE, AUTHOR)>
  <!ELEMENT TITLE (#PCDATA)>
  <!-- Autoreninformationen -->
  <!ELEMENT AUTHOR (#PCDATA)>
  <!ATTLIST AUTHOR Nationality CDATA 'Amerikaner'>
]
>
```

## Entities Definieren Und Verwenden

### Ein **Externes** Geparstes Parameter Entity Deklarieren

- Format

- `<!ENTITY % EntityName SYSTEM SystemLiteral>`

- *EntityName*

- Name der Entity

- *SystemLiteral*

- Systemliteral, das den Ort der Datei mit den Entity-Daten beschreibt.
- Das Systemliteral ist entweder in ‘ oder in “ eingeschlossen
- Das Systemliteral definiert die URI der Datei, welche die Entity-Daten enthält.
- Beispiel
  - Relativ adressiert  
`<!ENTITY % declarations SYSTEM „Declarations.dtd“>` oder
  - Absolut adressiert  
`<!ENTITY % declarations SYSTEM „http://ikl.com/Declarations.dtd“>`

# Entities Definieren Und Verwenden

## Ein **Externes** Geparstes Parameter Entity Deklarieren

- Einsatz

- Zusammenfassen von Definitionen bestimmter Artikelgruppen, z.B. CD's, Videos, ...

- Beispiel

```
<?xml version="1.0"?>
```

```
<!DOCTYPE INVENTORY
```

```
[
```

```
<!ELEMENT INVENTORY (BOOK | CD)*>
```

```
<!ENTITY % book_decls SYSTEM "BookExtern.dtd">
```

```
<!ENTITY % cd_decls SYSTEM "CDEExtern.dtd">
```

```
%book_decls;
```

```
%cd_decls;
```

```
]
```

```
>
```

```
<INVENTORY>
```

```
<BOOK>....
```



## Entities Definieren Und Verwenden

### Ein **Externes** Geparstes Parameter Entity Deklarieren

#### – CDExtern.dtd

```
<!ELEMENT CD (CDTITLE, COMPOSER, LENGTH)>
```

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

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

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

#### – BookExtern.dtd

```
<!ELEMENT BOOK (BOOKTITLE, AUTHOR, PAGES)>
```

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

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

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

## Entities Definieren Und Verwenden

### Entity-Referenzen Einfügen - Übersicht

- Allgemeine Referenzen
  - *&EntityName*
- Parameter-Referenzen
  - *%EntityName*
  - *EntityName*
    - Name, der dem Entity in der Deklaration zugewiesen wird.
- Ausnahme
  - Allgemeines externes nicht geparstes Entity

# Entities Definieren Und Verwenden

## Entity-Referenzen Einfügen - Übersicht

| <b>Entity-Typ</b>                        | <b>Format der Entity-Referenz, wobei <i>EntityName</i> den Namen des Entity bezeichnet</b>               | <b>Orte, an denen Sie eine Entity-Referenz einfügen (Beispiel)</b>   |
|--|--|--|
| Allgemein<br>intern<br>geparsed          | <i>&amp;EntityName;</i>  | Im Elementinhalt<br>Im Attributwert (als voreingestellter Wert in Attributdeklarationen oder im Start-Tag eines Elements)<br>Im Wert einer internen Entity-Deklaration |
| Allgemein<br>extern<br>geparsed          | <i>&amp;EntityName</i>   | Im Elementinhalt<br>Im Wert einer internen Entity-Deklaration  |
| Allgemein<br>extern<br>nicht<br>geparsed | <i>EntAttr=,EntityName'</i><br>wobei <i>EntAttr</i> ein Attribut vom Typ ENTITY oder ENTITIES bezeichnet | Eine Referenz auf diesen Entity-Typ können Sie nicht einfügen. Es ist aber zulässig, den Entity-Namen einem Attribut vom Typ ENTITY oder ENTITIES zuzuweisen           |

# Entities Definieren Und Verwenden

## Entity-Referenzen Einfügen - Übersicht

| Entity-Typ                | Format der Entity-Referenz, wobei <i>EntityName</i> den Namen des Entity bezeichnet   | Orte, an denen Sie eine Entity-Referenz einfügen (Beispiel)   |
|---------------------------|---|---|
| Parameter intern geparsed | % <i>EntityName</i> ;   | In einer DTD an einer für Markup geeigneten Stelle, nicht <i>innerhalb</i> Markup- Deklarationen  |
| Parameter extern geparsed | % <i>EntityName</i>   | In einer DTD an einer für Markup geeigneten Stelle, nicht <i>innerhalb</i> Markup- Deklarationen  |
| Zeichenreferenz           | &#9; oder &#xh;<br>9 bezeichnet den numerischen Code des Zeichens in Dezimaldarstellung,<br>h dagegen in Hexadezimaldarstellung | Im Elementinhalt<br>Im Attributwert (als Vorgabe in einer Attributlisten- Deklaration oder im Start-Tag eines Elements).<br>Im Wert einer internen Entity- Deklaration. |

## Entities Definieren Und Verwenden

### Entity-Referenzen Einfügen – Beispiel 1

- Allgemeine interne geparste Entities
  - Definition zweier Entities
    - *am, en*
  - Zuweisen eines Wertes (diese Entity)
    - an das Attribut *Nationality*
  - Beispieldatei
    - InventoryMitEntities.xml

# Entities Definieren Und Verwenden

## Entity-Referenzen Einfügen – Beispiel 1

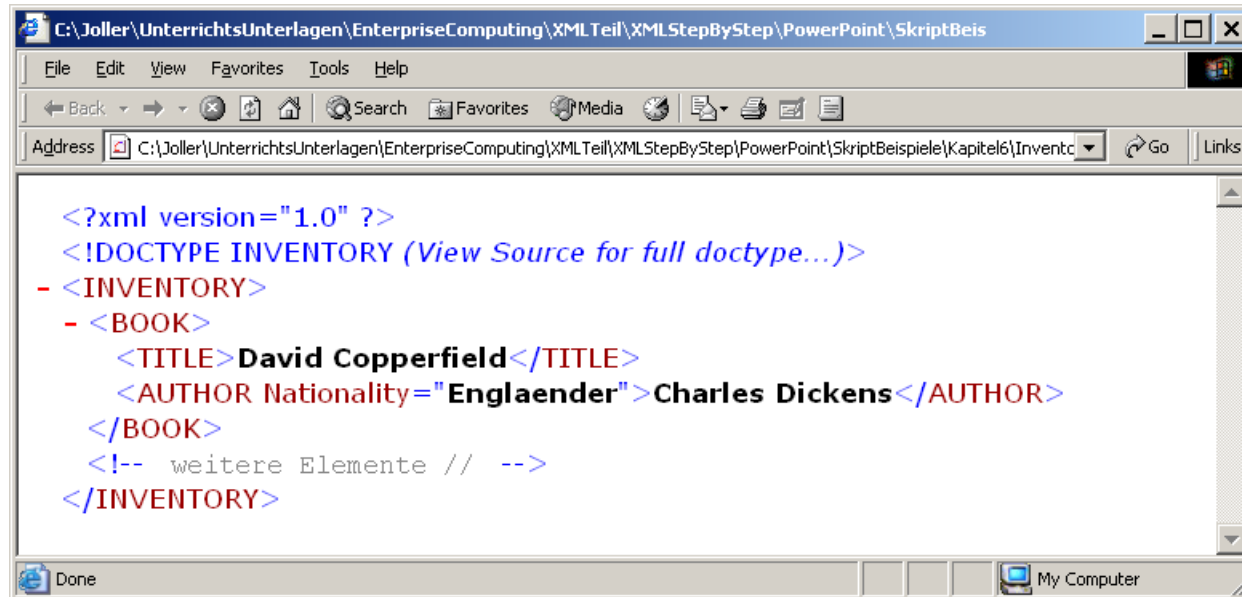
```
<?xml version="1.0"?>

<!DOCTYPE INVENTORY
[
  <!ENTITY am "Amerikaner">
  <!ENTITY en "Englaender">

  <!ELEMENT INVENTORY (BOOK*)>
  <!ELEMENT BOOK (TITLE, AUTHOR)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT AUTHOR (#PCDATA)>
  <!ATTLIST AUTHOR Nationality CDATA "&am;">
]
>
<INVENTORY>
  <BOOK>
    <TITLE>David Copperfield</TITLE>
    <AUTHOR Nationality="&en;">Charles Dickens</AUTHOR>
  </BOOK>
  <!-- weitere Elemente //-->
</INVENTORY>
```

# Entities Definieren Und Verwenden

## Entity-Referenzen Einfügen – Beispiel 1



The screenshot shows a web browser window with the following content:

```
<?xml version="1.0" ?>
<!DOCTYPE INVENTORY (View Source for full doctype...)>
- <INVENTORY>
  - <BOOK>
    <TITLE>David Copperfield</TITLE>
    <AUTHOR Nationality="Englaender">Charles Dickens</AUTHOR>
  </BOOK>
  <!-- weitere Elemente // -->
</INVENTORY>
```

## Entities Definieren Und Verwenden

### Entity-Referenzen Einfügen – Beispiel 2

- Allgemeine interne geparste Entities plus ein allgemeines externes geparstes Entity plus ein internes geparstes Entity, in das die beiden ersten eingefügt werden
  - Definition der Entities
    - *int\_entity, ext\_entity, combo\_entity*
  - Beispieldatei
    - `InventoryMitMixedEntities.xml`



# Entities Definieren Und Verwenden

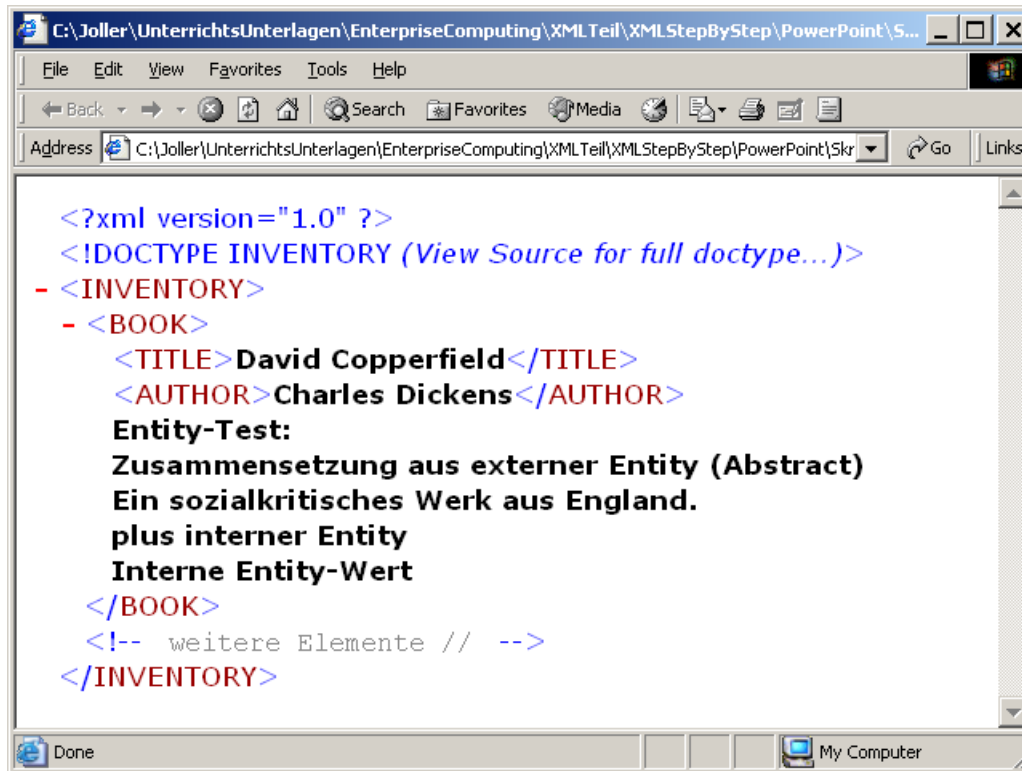
## Entity-Referenzen Einfügen – Beispiel 2

```
<?xml version="1.0"?>

<!DOCTYPE INVENTORY
  [
    <!ENTITY int_entity "Interne Entity-Wert">
    <!ENTITY ext_entity SYSTEM "Abstract.xml">
    <!ENTITY combo_entity "Zusammensetzung aus externer Entity (Abstract) &ext_entity;
plus interner Entity &int_entity;">
    <!ELEMENT INVENTORY (BOOK*)>
    <!ELEMENT BOOK (TITLE, AUTHOR)>
    <!ELEMENT TITLE (#PCDATA)>
    <!ELEMENT AUTHOR (#PCDATA)>
  ]
>
<INVENTORY>
  <BOOK>
    <TITLE>David Copperfield</TITLE>
    <AUTHOR>Charles Dickens</AUTHOR>
    Entity-Test: &combo_entity;
  </BOOK>
</INVENTORY>
```

# Entities Definieren Und Verwenden

## Entity-Referenzen Einfügen – Beispiel 2



The screenshot shows a web browser window with the following content:

```
<?xml version="1.0" ?>
<!DOCTYPE INVENTORY (View Source for full doctype...)>
- <INVENTORY>
- <BOOK>
  <TITLE>David Copperfield</TITLE>
  <AUTHOR>Charles Dickens</AUTHOR>
Entity-Test:
Zusammensetzung aus externer Entity (Abstract)
Ein sozialkritisches Werk aus England.
plus interner Entity
Interne Entity-Wert
</BOOK>
<!-- weitere Elemente // -->
</INVENTORY>
```

The rendered output of the XML is displayed below the code, showing the book title and author, followed by the entity test text.

## Entities Definieren Und Verwenden

### Zeichenreferenzen Einfügen

- Einsatz
  - Einfügen von beliebigen ISO/IEC Zeichen
- Format
  - `&#9;` einfügen eines ISO Zeichens
  - `&#xh;` einfügen des ISO Zeichens gemäss Hex Code
- Beispiel
  - `&#228` ä

# Entities Definieren Und Verwenden

## Zeichenreferenzen Einfügen

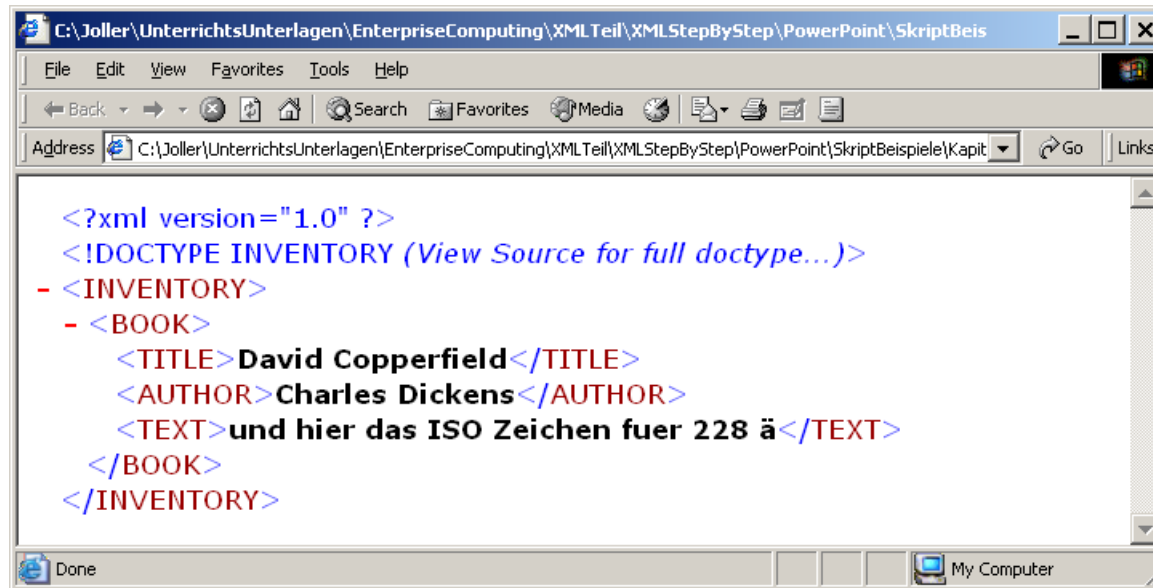
```
<?xml version="1.0"?>
```

```
<!DOCTYPE INVENTORY  
  [  
    <!ELEMENT INVENTORY (BOOK*)>  
    <!ELEMENT BOOK (TITLE, AUTHOR, TEXT)>  
    <!ELEMENT TITLE (#PCDATA)>  
    <!ELEMENT AUTHOR (#PCDATA)>  
    <!ELEMENT TEXT (#PCDATA)>  
  ]  
>
```

```
<INVENTORY>  
  <BOOK>  
    <TITLE>David Copperfield</TITLE>  
    <AUTHOR>Charles Dickens</AUTHOR>  
    <TEXT>und hier das ISO Zeichen fuer 228 &#228;</TEXT>  
  </BOOK>  
</INVENTORY>
```

# Entities Definieren Und Verwenden

## Zeichenreferenzen Einfügen



The screenshot shows a web browser window with the following XML code displayed in the main content area:

```
<?xml version="1.0" ?>
<!DOCTYPE INVENTORY (View Source for full doctype...)>
- <INVENTORY>
- <BOOK>
  <TITLE>David Copperfield</TITLE>
  <AUTHOR>Charles Dickens</AUTHOR>
  <TEXT>und hier das ISO Zeichen fuer 228 ä</TEXT>
</BOOK>
</INVENTORY>
```

The browser's address bar shows the path: C:\Joller\UnterrichtsUnterlagen\EnterpriseComputing\XMLTeil\XMLStepByStep\PowerPoint\SkriptBeispiele\Kapit... The status bar at the bottom indicates 'Done' and 'My Computer'.

## Entities Definieren Und Verwenden

### Vordefinierte Entities Verwenden

- Damit lassen sich auch die Zeichen %, &, <, > ... einfügen:

| <b>Vordefinierte Entity-Referenz</b> | <b>Eingefügtes Zeichen</b> | <b>Äquivalente Zeichenreferenz</b> |
|--------------------------------------|----------------------------|------------------------------------|
| &amp;                                | &                          | &#38;                              |
| &lt;                                 | <                          | &#60;                              |
| &gt;                                 | >                          | &#62;                              |
| &apos;                               | '                          | &#39;                              |
| &quot;                               | “                          | &#34;                              |

# Entities Definieren Und Verwenden

## Vordefinierte Entities Verwenden

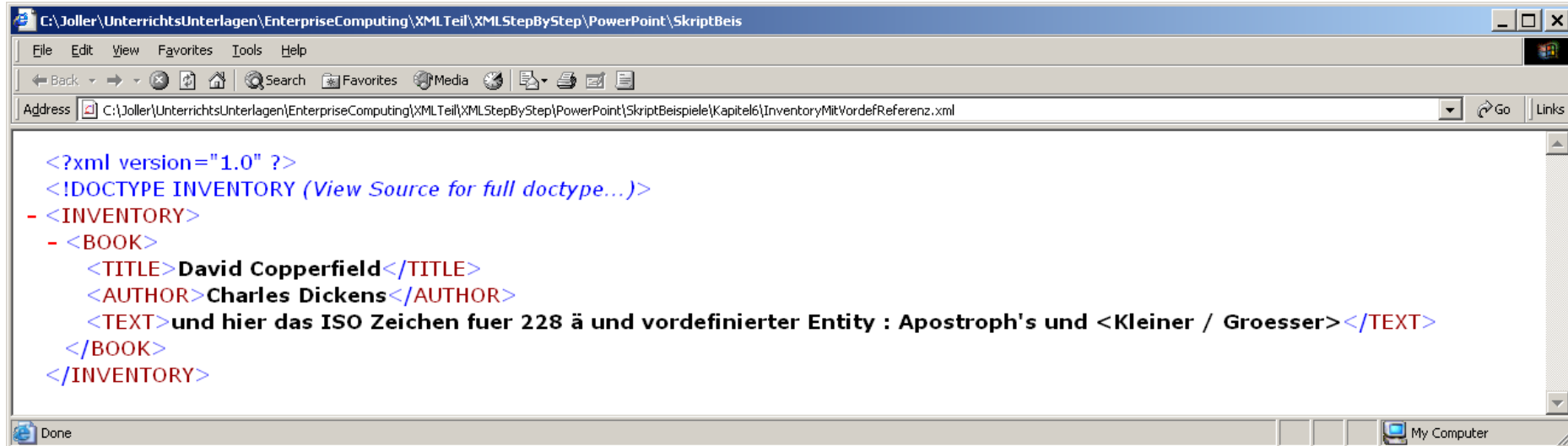
```
<?xml version="1.0"?>
```

```
<!DOCTYPE INVENTORY  
  [  
    <!ELEMENT INVENTORY (BOOK*)>  
    <!ELEMENT BOOK (TITLE, AUTHOR, TEXT)>  
    <!ELEMENT TITLE (#PCDATA)>  
    <!ELEMENT AUTHOR (#PCDATA)>  
    <!ELEMENT TEXT (#PCDATA)>  
  ]  
>
```

```
<INVENTORY>  
  <BOOK>  
    <TITLE>David Copperfield</TITLE>  
    <AUTHOR>Charles Dickens</AUTHOR>  
    <TEXT>und hier das ISO Zeichen fuer 228 &#228; und vordefinierter Entity :  
    Apostroph&apos;s und &lt;Kleiner / Groesser&gt;</TEXT>  
  </BOOK>  
</INVENTORY>
```

# Entities Definieren Und Verwenden

## Vordefinierte Entities Verwenden



The screenshot shows a web browser window with the following content:

```
<?xml version="1.0" ?>
<!DOCTYPE INVENTORY (View Source for full doctype...)>
- <INVENTORY>
- <BOOK>
  <TITLE>David Copperfield</TITLE>
  <AUTHOR>Charles Dickens</AUTHOR>
  <TEXT>und hier das ISO Zeichen fuer 228 ä und vordefinierter Entity : Apostroph's und <Kleiner / Groesser></TEXT>
</BOOK>
</INVENTORY>
```

The browser's address bar shows the file path: C:\Joller\Unterrichtsunterlagen\EnterpriseComputing\XMLTeil\XMLStepByStep\PowerPoint\SkriptBeispiele\Kapitel6\InventoryMitVordefReferenz.xml. The status bar at the bottom indicates 'Done' and 'My Computer'.



## Entities Definieren Und Verwenden

### Vordefinierte Entities Verwenden - Standalone

- Falls keine externen DTDs benötigt werden, kann man die standalone Option der XML Anweisung auf „yes“ setzen:
  - `<?xml version="1.0" standalone="yes">`
  - Damit werden keine externen Dateien verarbeitet.
  - Der Parser muss also alle Definitionen aus dem Dokument erkennen und auswerten können.

## Entities Definieren Und Verwenden

### Entities Einem Dokument Hinzufügen

- Übung

- Fügen Sie einige allgemeine Entities in das Beispieldokument *Inventory Valid.xml* aus dem vorherigen Kapitel

- Öffnen Sie das Dokument

- Ergänzen Sie das Dokument durch den Text aus der Datei *Delta.xml*

- Delta.xml enthält

- » Entities (allgemein geparste Entities); diese können Sie in BINDING-Elemente einfügen.

- Damit wird Ihre Beschreibung konsistenter, einheitlicher.

- » Die nächsten Entities enthalten externe nicht geparste Entities, die auf die Dateien der Buchbesprechungen verweisen.

# Entities Definieren Und Verwenden

## Entities Einem Dokument Hinzufügen

- Übung

- Ergänzen Sie die Attributliste  
<!ATTLIST BOOK InStock (ja|nein) #REQUIRED  
**Review ENTITIES #IMPLIED**>  
*Reviews* wird damit ein optionales Attribut
- Ersetzen Sie die BINDING Angaben durch die Entities  
<BINDING>Taschenbuch</BINDING> durch  
<BINDING>&trade;</BINDING>  
(; am Ende von &trade; nicht vergessen)  
<BINDING>Gebundene Ausgabe</BINDING> durch  
<BINDING>&hard</BINDING>

## Entities Definieren Und Verwenden

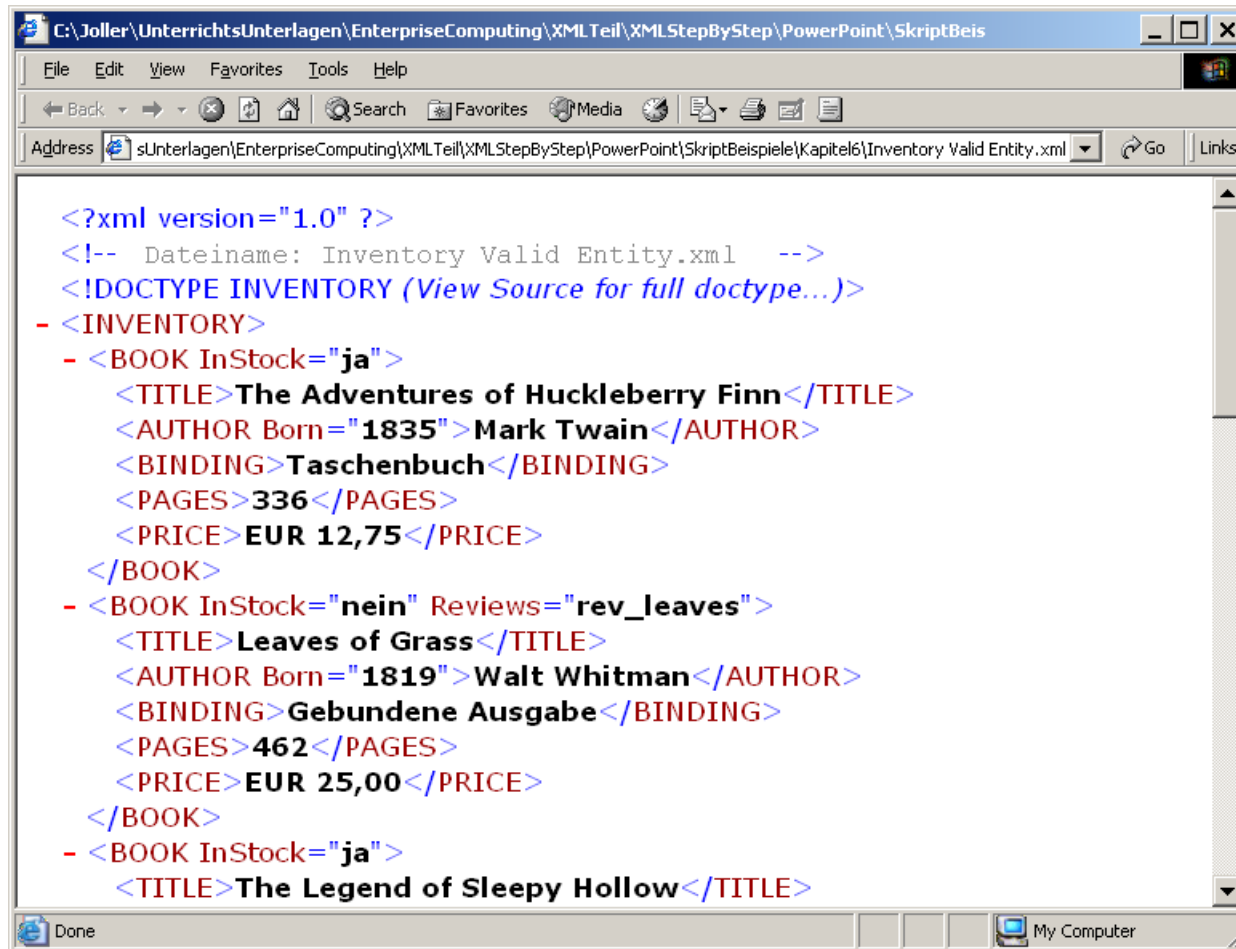
### Entities Einem Dokument Hinzufügen

- Übung

- Ergänzen Sie die Attribute *Reviews* im BOOK-Element
  - Für “Leaves of Grass”  
<BOOK InStock=“nein” Reviews=“rev\_leaves”>
  - Für “Der Graf von Monte Christo”  
<BOOK InStock=“ja” Review=“rev\_christo1 rev\_christo2”>
  - Für “Harry Potter und der Stein der Weisen”  
<BOOK InStock=“nein” Reviews=“rev\_potter”>
- Ändern Sie den Namen der Datei (auch im Kommentar im Header) auf “Inventory Valid Entity.xml”
- Speichern Sie die Datei und testen Sie Ihr Werk
  - Eine korrekte Lösung ist auf dem Server / der CD

# Entities Definieren Und Verwenden

## Entities Einem Dokument Hinzufügen



The screenshot shows a web browser window with the address bar displaying the file path: `sUnterlagen\EnterpriseComputing\XMLTeil\XMLStepByStep\PowerPoint\SkriptBeispiele\Kapitel6\Inventory Valid Entity.xml`. The browser's status bar at the bottom shows "Done" and "My Computer". The main content area displays the following XML code:

```
<?xml version="1.0" ?>
<!-- Dateiname: Inventory Valid Entity.xml -->
<!DOCTYPE INVENTORY (View Source for full doctype...)>
- <INVENTORY>
- <BOOK InStock="ja">
  <TITLE>The Adventures of Huckleberry Finn</TITLE>
  <AUTHOR Born="1835">Mark Twain</AUTHOR>
  <BINDING>Taschenbuch</BINDING>
  <PAGES>336</PAGES>
  <PRICE>EUR 12,75</PRICE>
</BOOK>
- <BOOK InStock="nein" Reviews="rev_leaves">
  <TITLE>Leaves of Grass</TITLE>
  <AUTHOR Born="1819">Walt Whitman</AUTHOR>
  <BINDING>Gebundene Ausgabe</BINDING>
  <PAGES>462</PAGES>
  <PRICE>EUR 25,00</PRICE>
</BOOK>
- <BOOK InStock="ja">
  <TITLE>The Legend of Sleepy Hollow</TITLE>
```