XML – Eine Einführung
XML Dokumente Erstellen

Gültige XML Dokumente Erstellen

- Nach dem Durcharbeiten dieser Lektion sollten Sie in der Lage sein
  - Zu definieren, was man unter einem gültigen Dokument versteht
  - Zu wissen welche Vorteile gültige Dokumente haben
  - einem Dokument eine Dokumenten-Typ Definition hinzufügen zu können
  - Elemente und Attribute für gültige Dokumente definieren können
  - Ein gegebenes Dokument in XML in ein gültiges Dokument zu transformieren

Gültige XML Dokumente Erstellen
Die Grundkriterien Für Ein Gültiges XML Dokument

- Jedes XML Dokument muss wohlgeformt sein, also den minimalen Anforderungen von weiter vorne genügen.
  - Falls dies nicht der Fall ist, wird ein Dokument nicht als XML Dokument angesehen.
- Ein wohlgeformtes Dokument kann ausserdem *gültig* sein.
  - In diesem fall genügt das wohlgeformte Dokument den folgenden zwei zusätzlichen Anforderungen
    - Der Prolog des Dokuments muss eine Dokumenttyp Deklaration enthalten, die wiederum eine *Dokumententyp-Definition* (DTD) zur Definition der Dokumentenstruktur enthält.
    - Das restliche Dokument muss der in der DTD definierten Struktur entsprechen

- Wohlgeformtheit wird durch eine in der XML Spezifikation definierte Regelmenge festgelegt. Diesen Regeln muss ein wohlgeformtes XML Dokument genügen.
  - Jeder Verstoss gegen eine dieser Grundregeln wird als fehler angesehen.Der XML Prozessor muss diesen Verstoss als fehler anzeigen.

- Die Struktur gültiger XML Dokumente wird durch eine DTD (oder wie wir später noch sehen werden, dem XML Schema) definiert.
  - Die Erstellung gültiger XML Dokumente scheint mit viel Aufwand verbunden zu sein:
    - Zuerst wird eine XML Dokumenttyp Beschreibung erstellt
    - Dann wird ein XML Dokument geschrieben
  - Falls Sie aber viele Dokumente standardisieren wollen, bietet sich ein verfahren mit einer generellen Definition plus Instanzen geradezu an
    - Die generelle Beschreibung garantiert Einheitlichkeit
    - Die Instanz, das konkrete Dokument sieht so aus, wie es von irgend einer zentralen Stelle gefordert wird.
    - Die Überprüfung geschieht mit Hilfe eines XML Prozessors

- Gültige XML Dokumente sind nützlich
  - Wenn die Einheitlichkeit einer Gruppe gleichartiger Dokumente gewährleistet werden muss.
    - Der XML Standard definiert eine DTD als „eine Grammatik für eine Klasse von Dokumenten“.
  - Wenn die Dokumente von benutzerspezifischer Software verarbeitet werden, die eine bestimmte Dokumentenstruktur erwartet.
    - Die Dokumente müssen von der Verarbeitungssoftware erkannt werden.
    - Beispiel: Internet Explorer
      - XML Dokumente werden beim Öffnen in IE geprüft (MSXML Parser).

- Aufbau einer DTD
  - Block mit XML Markupcode im Prolog der XML Deklaration.

Prolog

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

Dokumenttyp Deklaration entweder vor oder nach dem Kommentar

```
<!- DTD...-->
<!-- Dateiname: Inventory.xml -->
<!- DTD ...-->
```

Dokumentelement

```
<INVENTORY>...</INVENTORY>
```

- Eine Dokumenttyp Deklaration besitzt das folgende allgemeine Format

  `<!DOCTYPE` *`Name DTD`*`>`

  - *Name* : gibt den Namen des Dokumentelements an. Der Name des eigentlichen Dokumentelements muss mit dem Namen, den Sie hier eingeben, exakt übereinstimmen.
    - Beispiel <!DOCTYPE INVENTORY *DTD*>
  - *DTD :* ist die eigentliche Dokumenttyp-Deklaration, welche Elemente, Attribute und sonstige Eigenschaften des Dokuments definiert. DOCTYPE wird gross geschrieben.

Gültige XML Dokumente Erstellen
Die DTD Erstellen

- Die DTD besteht aus einer linken eckigen Klammer ([), gefolgt von Markup-Deklarationen sowie einer rechten eckigen Klammer (]).
- Markup Deklarationen:
  - Beschreiben die logische Struktur des Dokuments
    Elemente, Attribute und weitere Eigenschaften des Dokuments
  - Beispiel: das Dokument hat lediglich Elemente vom Typ SIMPLE

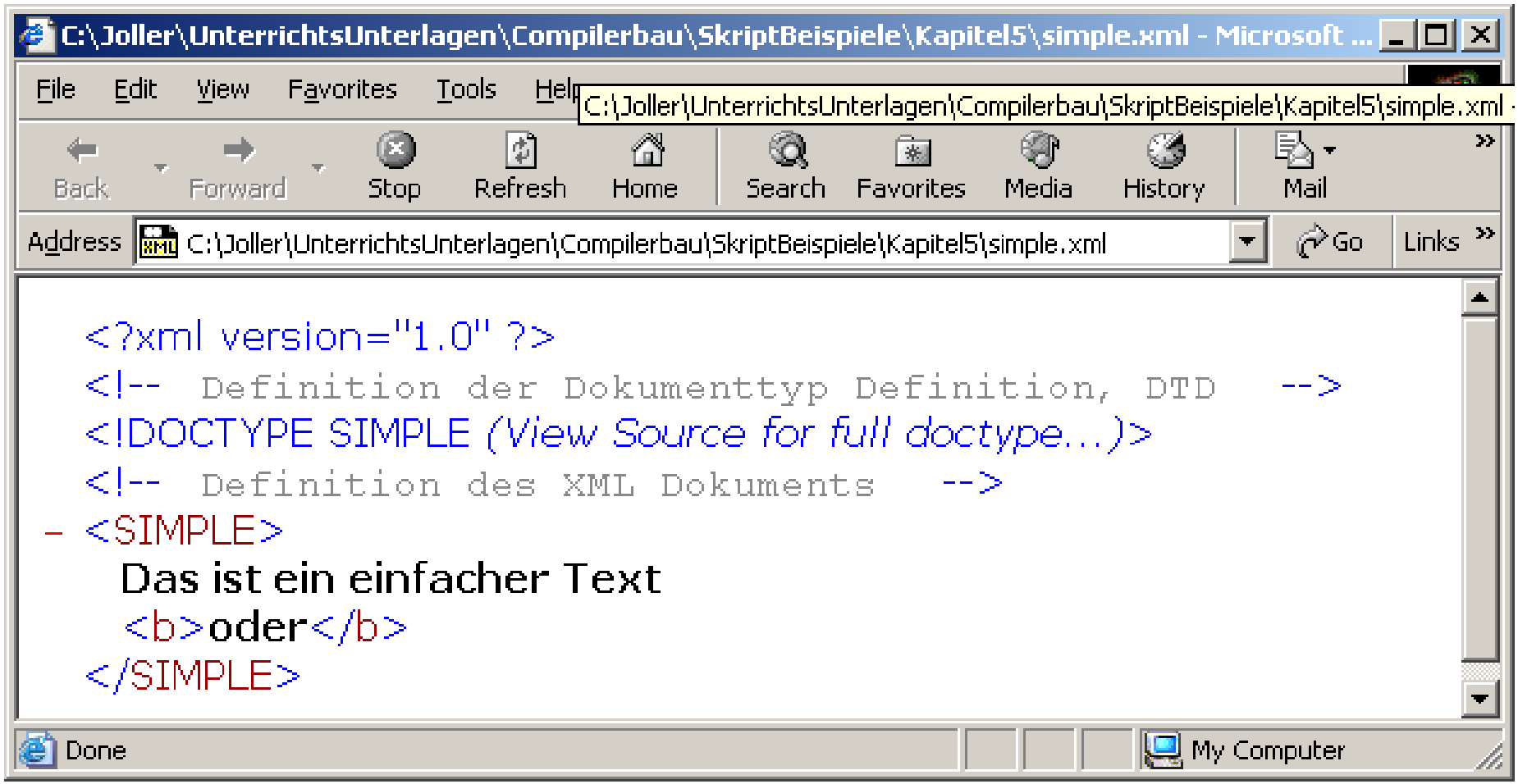
```
<?xml version="1.0"?>

<!-- Definition der Dokumenttyp Definition, DTD -->

<!DOCTYPE SIMPLE
    [
    <!ELEMENT Alles ANY>
    ]
>

<!-- Definition des XML Dokuments -->
<SIMPLE>Das ist ein einfacher Text <b>oder</b></SIMPLE>
```

Gültige XML Dokumente Erstellen
Die DTD Erstellen



- Eine DTD kann folgende Typen von Markup Deklarationen enthalten:
  - Elementtyp Deklarationen
    - Diese definieren den Typen der Elemente, die das Dokument enthalten darf, sowie den Inhalt und die Reihenfolge der Elemente.
  - Attributlisten-Deklarationen
    - Jede Attributlisten-deklaration definiert die Namen der Attribute, die auf einen bestimmten Elementtyp angewendet werden, sowie die Datentypen und Vorgabewerte dieser Attribute.
  - Entity-Deklarationen
    - Sie können Entities verwenden, um häufig verwendete Textblöcke zu speichern oder nicht XML konforme Daten in Ihr Dokument aufzunehmen.

- Eine DTD kann folgende Typen von Markup Deklarationen enthalten (Fortsetzung):
  - Notationsdeklarationen
    - Eine Notation beschreibt ein Datenformat oder identifiziert das Programm, das ein bestimmtes Format verarbeitet.
  - Verarbeitungsanweisung
    - Verarbeitungsanweisungen werden später erläutert.
  - Kommentare
    - Kommentare werden später besprochen
  - Parameter-Entity-Referenzen
    - Jedes Element aus dieser Liste kann in einem Parameter Entity enthalten sein und über eine Parameter-Entity-Referenz eingefügt werden.

- In einem gültigen XML Dokument müssen Sie die Typen aller Elemente, die Sie im Dokument verwenden, explizit in einer *Elementtyp-Deklaration* innerhalb der DTD deklarieren.
  - Eine Elementtyp-Deklaration beschreibt den Namen des Elementtyps und den erlaubten Inhalt des Elements
    - Zusammen bilden die Elementtyp-Deklarationen in der DTD, ähnlich einem Datenbankentwurf, die gesamte logische Struktur des Dokuments ab: die Elementtypen, deren reihenfolge und die Inhaltsspezifikation der Elemente.

- Eine Elementtyp-Deklaration besitzt das folgende allgemeine Format:

  <!ELEMENT *Name Inhaltsspezifikation*>

  – *Name*: bezeichnet den Namen des Elementtyps, der deklariert wird
  – *Inhaltsspezifikation*: ist die Spezifikation, die den zulässigen Inhalt eines Elements definiert.
  – Beispiele:

    <!ELEMENT TITLE (#PCDATA)>

    Deklaration eines Elementtyps TITLE, der nur Zeichendaten enthalten darf

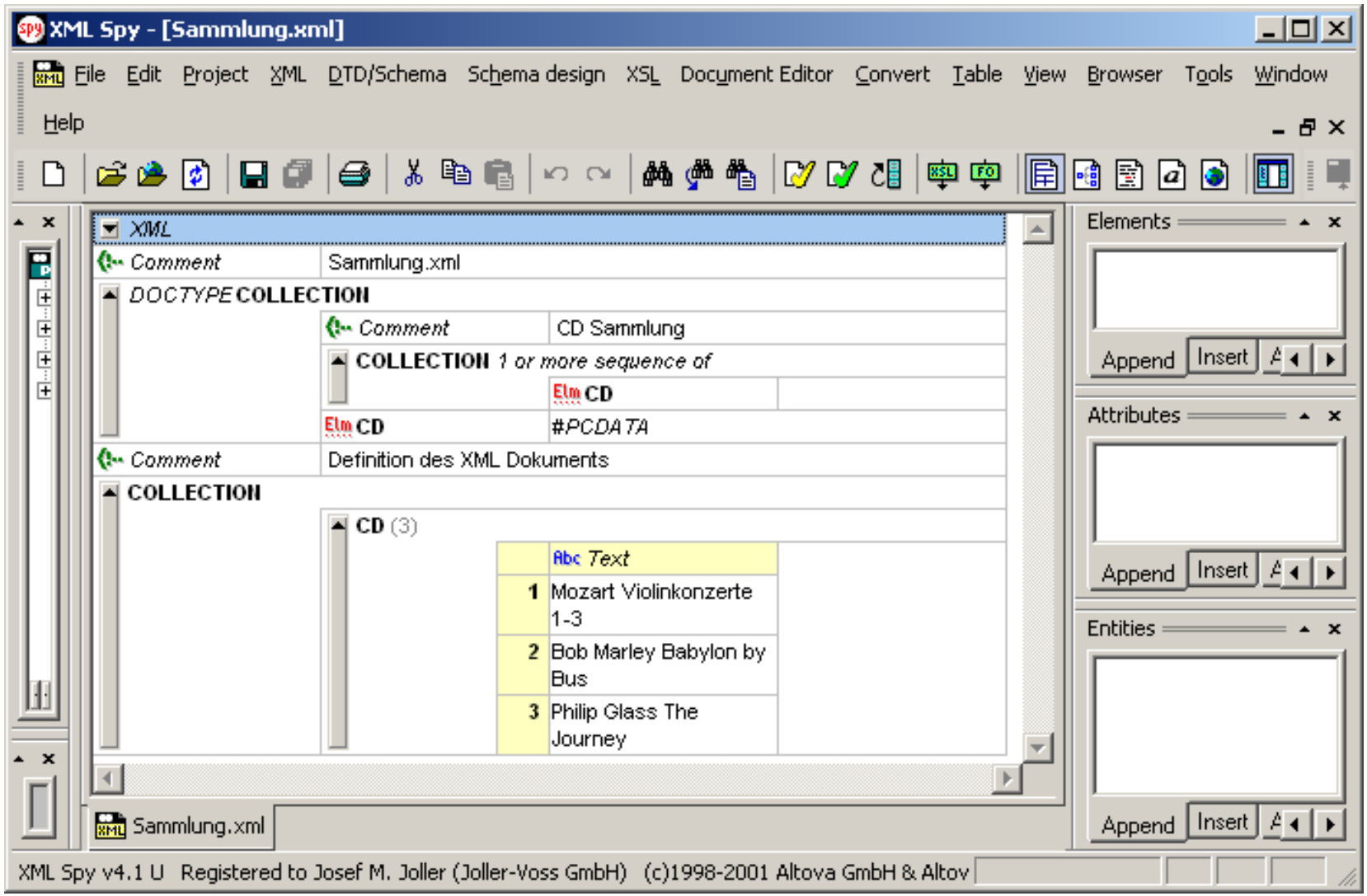
    <!ELEMENT ALLGEMEIN ANY>

    Deklaration eines Elementtyps ALLGEMEIN, der jeden Inhaltstyp enthalten darf

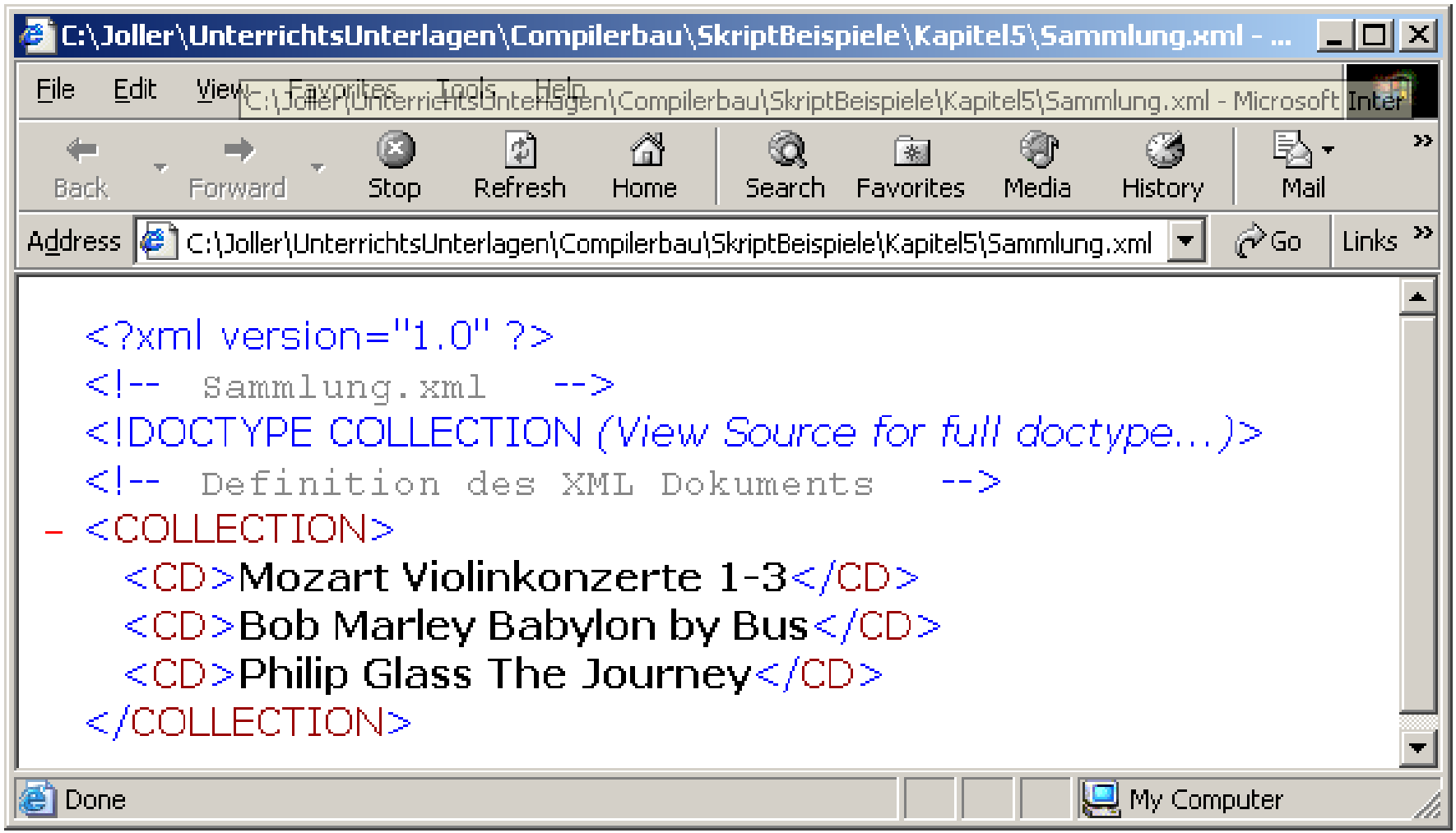
Gültige XML Dokumente Erstellen
Das Format Einer Elementtyp-Deklaration

```
<?xml version="1.0"?>
<!-- Sammlung.xml -->
<!DOCTYPE COLLECTION
        [
        <!-- CD Sammlung -->
        <!ELEMENT COLLECTION (CD)+>
        <!ELEMENT CD (#PCDATA)>
        ]
>
<!-- Definition des XML Dokuments -->
<COLLECTION>
        <CD>Mozart Violinkonzerte 1-3</CD>
        <CD>Bob Marley Babylon by Bus</CD>
        <CD>Philip Glass The Journey</CD>
</COLLECTION>
```

Gültige XML Dokumente Erstellen
Das Format Einer Elementtyp-Deklaration



Gültige XML Dokumente Erstellen
Das Format Einer Elementtyp-Deklaration



Gültige XML Dokumente Erstellen
Die Elementinhaltspezifikation

- Sie können den Inhalt des Elements (die Inhaltsspezifikation) auf vier unterschiedliche Arten festlegen
  - EMPTY, leeres Element
    - Mit dem Schlüsselelement EMPTY zeigen Sie an, dass das Element keinen Inhalt besitzen darf.
    - Beispiel:
      DTD Teil: <!ELEMENT BILD EMPTY>
      XML Teil: <IMAGE></IMAGE>
      <IMAGE/>
  - ANY, beliebiger Inhalt
    - Beispiel:
      DTD : <!ELEMENT Diverses ANY>
      XML : <Diverses>Das ? : oder das?</Diverses>
  - Elementinhalt , untergeordnete Elemente
  - Gemischter Inhalt

Gültige XML Dokumente Erstellen
Elementinhalte Spezifizieren

- Besitzt ein Element den Inhaltstyp ‚Elementinhalt', dann darf es selbst nur die angegebenen untergeordneten Elemente, nicht jedoch Zeichendaten enthalten.
- Beispiel:

```
<?xml version="1.0"?>
<!-- Buch.xml //-->
<!DOCTYPE BUCH
    [
    <!ELEMENT Buch (Titel, Autor)>          <!—Inhaltsmodell //-->
    <!ELEMENT Titel (#PCDATA)>
    <!ELEMENT Autor (#PCDATA)>
    ]
>
<Buch>
    <Titel>Das Universum in der Nussschale</Titel>
    <Autor>Stephen Hawking</Autor>
</Buch>
```

Gültige XML Dokumente Erstellen
Elementinhalte Spezifizieren

- Das Inhaltsmodell kann eines der beiden Formate besitzen:
  - **Sequenz** : Dieses Format eines Inhaltsmodells zeigt an, dass das Element eine bestimmte Reihenfolge untergeordneter Elemente besitzt. Die Namen der untergeordneten Elementtypen werden durch Kommas getrennt.

```
<?xml version="1.0"?>
<!-- Berg.xml //-->
<!DOCTYPE Berg
   [
   <!ELEMENT Berg (Name, Hoehe, Land)>
   <!ELEMENT Name (#PCDATA)>
   <!ELEMENT Hoehe (#PCDATA)>
   <!ELEMENT Land (#PCDATA)>
   ]
>

<Berg>
   <Name>Wheeler</Name>
   <Hoehe>13161</Hoehe>
   <Land>New Mexico</Land>
</Berg>
```

Gültige XML Dokumente Erstellen
Elementinhalte Spezifizieren

- Das Inhaltsmodell kann eines der beiden Formate besitzen (2):
  - **Auswahl** : Dieses Format eines Inhaltsmodells zeigt an, dass das Element aus einer Folge möglicher untergeordneter Elemente besitzen darf..

```
<?xml version="1.0"?>
<!-- Film.xml //-->
<!DOCTYPE FILM
   [
   <!ELEMENT FILM (STAR|NARRATOR|INSTRUCTOR)> <!– Auswahl //→
   <!ELEMENT STAR (#PCDATA)>
   <!ELEMENT NARRATOR (#PCDATA)>
   <!ELEMENT INSTRUCTOR (#PCDATA)>
   ]
>

<FILM>
   <STAR>Robert Redford</STAR>
</FILM>
```

Gültige XML Dokumente Erstellen
Elementinhalte Spezifizieren

- Inhaltsmodell
  - ? Keines oder das vorangehende Element
  - \+ Eines oder mehrere der vorausgehenden Elemente
  - Keines oder mehrere der vorausgehenden Elemente

  - Beispiel
    - <!ELEMENT BERG (**NAME+, HOEHE?, LAND**)]>
    - <BERG>

```
        <NAME>Pueblo Peak</NAME>
        <NAME>Taos Mountain</NAME>
        <LAND>New Mexico</LAND>
</BERG>
```

- Hat ein Element einen *gemischten Inhalt*, dann kann es Zeichendaten enthalten.Die untergeordneten Elemente können in beliebiger Reihenfolge und in beliebiger Anzahl oder gar nicht enthalten.
  - Sie können also die Typen der untergeordneten Elemente erzwingen, nicht jedoch die Reihenfolge oder die Anzahl eines untergeordneten Elementtyps und Sie können auch nicht vorschreiben, dass ein bestimmter untergeordneter Elementtyp vorkommen muss.

Gültige XML Dokumente Erstellen
Gemischten Inhalt Spezifizieren

- Um einen Elementtyp mit gemischtem Inhalt zu deklarieren, verwenden Sie eines der beiden folgenden Formate eines Inhaltsmodells:
  - **Nur Zeichendaten**
    - Beispiel:
      - DTD: <!ELEMENT UNTERTITEL (#PCDATA)>
      - XML: <UNTERTITEL>Ein neuer Ansatz</UNTERTITEL>
        oder <UNTERTITEL></UNTERTITEL>
    - Hinweis:
      - #PCDATA steht für Parsed Character Data; der XML Parser prüft die Zeichenkette; daher dürfen keine < oder & darin vorkommen
  - **Zeichendaten plus optionale untergeordnete Elemente**
    - Beispiel:
      - DTD: <!ELEMENT TITEL (#PCDATA | UNTERTITEL)>
      - XML: <TITEL>Moby Dick <UNTERTITEL>oder der Wal</UNTERTITEL></TITEL>

Gültige XML Dokumente Erstellen
Attribute Deklarieren

- In einem gültigen XML-Dokument müssen Sie ausserdem alle Attribute explizit deklarieren, die Sie zusammen mit den Elementen des Dokuments verwenden wollen. Sie definieren alle einem bestimmten Element zugeordneten Attribute über einen DTD Typ, die so genannte *Attributlisten-Deklaration*. Diese Deklaration erfüllt folgende Zwecke:
  - Sie definiert die Namen der Attribute, die dem Element zugeordnet sind.
  - Sie spezifiziert den Datentyp jedes Attributs
  - Sie gibt für jedes Attribut an, ob es erforderlich ist.

Gültige XML Dokumente Erstellen
Das Format Einer Attributlisten-Deklaration

- Eine Attributlisten Deklaration hat folgendes Format:
  - <!ATTLIST *Name AttDefs*>
  - *Name* bezeichnet den Typnamen des Elements, dem das Attribut bzw. die Attribute zugeordnet sind.
  - *AttDefs* besteht aus einer Folge von einer oder mehreren *Attributdefinitionen*, die jeweils ein Attribut definieren.

- Eine Attributdefinition besitzt folgende Syntax:
  - *Name AttTyp VorgabeDekl*
  - *Name* steht für den Namen des Attributs
  - *AttTyp* ist der *Attributtyp*, also der Typ des Wertes, der dem Attribut zugewiesen werden kann.
  - *VorgabeDekl* ist die *Vorgabedeklaration*. Sie zeigt an, ob das Attribut erforderlich ist und stellt weitere Informationen zur Verfügung.

Gültige XML Dokumente Erstellen
Das Format Einer Attributlisten-Deklaration

- Beispiel
  - <!ELEMENT FILM (TITLE , (STAR | NARRATOR | INSTRUCTOR))>
  - <!ATTLIST FILM Class CDATA „fictional“ Year CDATA #REQUIRED>

    Attribut Listen Deklaration | Name des Elem. | AttrName | Vorgabedekl. | AttrName | Vorgabedekl.

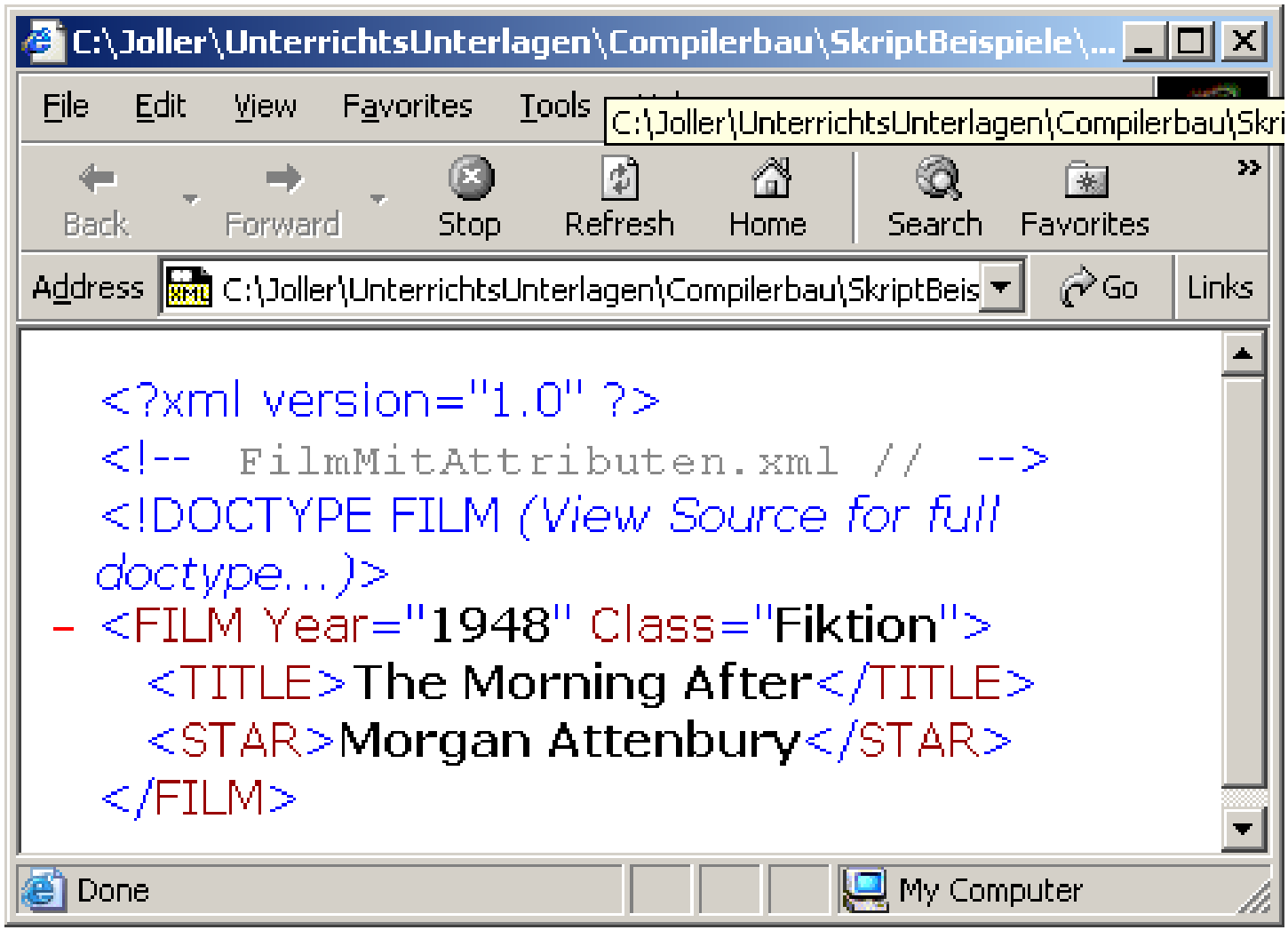
  - Sie können dem Attribut Class jede gültige Zeichenkette in Anführungszeichen (CDATA) zuweisen
  - Falls Sie es weglassen, wird dem Attribut automatisch der Vorgabewert „fictional“ zugewiesen.
  - Dem Attribut Year **muss** immer ein Wert (CDATA) zugewiesen werden

Gültige XML Dokumente Erstellen
Das Format Einer Attributlisten-Deklaration

```
<?xml version="1.0"?>
<!-- FilmMitAttributen.xml //-->
<!DOCTYPE FILM
    [
    <!ELEMENT FILM (TITLE, (STAR|NARRATOR|INSTRUCTOR))>
    <!ATTLIST FILM Class CDATA "Fiktion" Year CDATA #REQUIRED>
    <!ELEMENT TITLE (#PCDATA)>
    <!ELEMENT STAR (#PCDATA)>
    <!ELEMENT NARRATOR (#PCDATA)>
    <!ELEMENT INSTRUCTOR (#PCDATA)>
    ]
>

<FILM Year="1948">
    <TITLE>The Morning After</TITLE>
    <STAR>Morgan Attenbury</STAR>
</FILM>
```

Gültige XML Dokumente Erstellen
Das Format Einer Attributlisten-Deklaration



Gültige XML Dokumente Erstellen
Der Attributtyp

- Beim Attributtyp handelt es sich um die zweite erforderliche Komponente einer Attributdefinition
  - Der Attributtyp gibt an, welche Art von Wert Sie dem Attribut im Dokument zuweisen können.

  - **<!ATTLIST** Attributlisten Deklaration | **FILM** Name des zugehörigen Elements | **Class** Attribut-name | **CDATA** Attribut-typ | **„Fiktion“>** Vorgabe-deklaration

- Sie haben drei Möglichkeiten, den Attributnamen anzugeben:
  - **Zeichenkettentyp** : Zeichenkette in Anführungszeichen (Literal)
    - Beispiel: <!ATTLIST FILM Class **CDATA** „fictional“>
  - **Token-Typ**: erzwungene Werte (siehe weiter hinten)
  - **Aufzähltyp** : Wert aus einer Liste von Werten (s.w.h.)

Gültige XML Dokumente Erstellen
Einen Token-Typ Spezifizieren

- Attributwerte vom Token-Typ bestehen aus Zeichenketten in Anführungszeichen.
  - Zusätzlich muss der Wert konform zu der Einschränkung sein, die Sie mithilfe eines entsprechenden Schlüsselworts in der Attributdefinition spezifizieren, beispielsweise ID (ein eindeutiger Wert).

```
<?xml version="1.0"?>
<!-- ArtikelListe.xml //-->
<!DOCTYPE INVENTORY
    [
    <!ELEMENT INVENTORY (ITEM*)>
    <!ELEMENT ITEM (#PCDATA)>
    <!ATTLIST ITEM StockCode ID #REQUIRED>
    ]
>
<!-- jedes Item besitzt einen eigenen StockCode //-->
<INVENTORY>
    <ITEM StockCode="S0123">Teekanne mit Rosenmuster</ITEM>
    <ITEM StockCode="S0124">Elektrische Kaffeemuehle</ITEM>
    <ITEM StockCode="S0132">Backofen Thermometer</ITEM>
</INVENTORY>
```

- ID
  - Das Attribut muss in jedem Element einen eindeutigen Wert besitzen. Der Wert muss mit einem Buchstaben oder einem Unterstrich (_) beginnen, dem null oder mehrere Buchstaben, Zahlen, Punkte (.), Bindestriche (-) oder Unterstriche (_) folgen können.Falls ein Doppelpunkt verwendet wird, muss dieser der Definition des Namensraumes dienen.
  - Ein bestimmter Elementtyp darf nur ein Attribut vom Typ ID besitzen und die voreingestellte Deklaration des Attributs muss entweder #REQUIRED oder #IMPLIED lauten.

- IDREF
  - Der Attributwert muss mit dem Wert eines Attributs vom Typ ID in einem Element innerhalb des Dokuments übereinstimmen. Dieser Attributtyp verweist also auf den eindeutigen Bezeichner eines anderen Attributs.
    Beispielsweise können Sie ein IDREF Attribut namens *GoesWith* zum ITEM Element hinzufügen:

    - <!ELEMENT ITEM (#PCDATA)>
      <!ATTLIST ITEM StockCode ID #REQUIRED
      GoesWith IDREF #IMPLIED>
    - <ITEM StockCode=„ABB123“>Elektrische Kaffe-Mühle</ITEM>
      <ITEM StockCode=„Solis“ GoesWith=„ABB123“>
      Kaffee-Maschine
      </ITEM><!– umgekehrt wäre wahrscheinlich besser //-->

- IDREFS
  - Dieser Attributtyp entspricht dem Typ IDREF, doch der Wert kann Verweise auf verschiedene Bezeichner enthalten, die durch Leerzeichen getrennt werden und alle in der in Anführungszeichen eingeschlossenen Zeichenkette stehen
    Beispiel:
    - <!ATTLIST ITEM StockCode ID #REQUIRED GoesWith IDREFS #IMPLIED>
    - <ITEM StockCode=„S034“>Elektrische Kaffeemühle</ITEM>
      <ITEM StockCode=„S039“>1 Pfund Kaffeebohnen</ITEM>
      <ITEM StockCode=„S047“ GoesWith=„S034 S039“>Kaffeemühlen-Bürste</ITEM>

- ENTITY
  - Der Attributwert muss mit dem Namen eines in der DTD deklarierten nicht geparsten Entity übereinstimmen. Ein nicht geparstes Entity verweist auf eine externe Datei, normalerweise eine, die XML fremde Daten enthält.
    Beispiel:
    - <!ELEMENT IMAGE EMPTY>
      <!ATTLIST IMAGE Source ENTITY #REQUIRED>
    - <IMAGE Source=„Logo“> <!– Logo wurde deklariert //-->

- ENTITIES
  - Dieser Attributtyp entspricht dem ENTITY Typ, jedoch kann der Wert die Namen mehrerer nicht geparster Entities enthalten, die durch Leerzeichen voneinander getrennt und alle in der in Anführungszeichen eingeschlossenen Zeichenkette enthalten sind. Beispiel:
    - <!ELEMENT IMAGE EMPTY>
      <!ATTLIST IMAGE Source ENTITIES #REQUIRED>
    - <IMAGE Source=„LogoGif LogoBmp“>

- NMTOKEN
  - Der Wert ist ein Namenstoken, also ein Mane, der aus einer beliebigen Zusammenstellung von Buchstaben, Zahlen, Punkten, Bindestrichen oder Unterstrichen besteht. Ein Namenstoken kann ausserdem einen einzelnen Doppelpunkt enthalten, nicht jedoch an der ersten Zeichenposition.
    Beispiel:
    - <!ELEMENT BOOK (#PCDATA)>
      <!ATTLIST BOOK ISBN NMTOKEN #REQUIRED>
    - <BOOK ISBN=„09-99999-999-9“>Das Haus am See</BOOK>

- NMTOKENS
  - Dieser Attributtyp entspricht dem Typ NMTOKEN, jedoch kann der Wert mehrere Namenstoken umfassen, die durch Leerzeichen getrennt und alle innerhalb der in Anführungszeichen stehenden Zeichenkette enthalten sind.
    Beispiel:
    - <!ELEMENT SHIRT (#PCDATA)>
      <!ATTLIST SHIRT Codes NMTOKENS #REQUIRED>
    - <SHIRT Codes=„38 21 97“>Hemd mit langen Ärmeln</SHIRT>

Gültige XML Dokumente Erstellen
Einen Aufzählungstyp Spezifizieren

- Wie jeder Attributwert muss auch der Wert, den Sie einem Aufzählungstyp zuweisen, eine in Anführungszeichen eingeschlossene Zeichenkette sein. Zusätzlich muss der Wert mit einem der Namen übereinstimmen, die Sie in der Attributtyp-Spezifikation auflisten, die einem der folgenden Formate entsprechen muss:
  - Eine öffnende Klammer, gefolgt von einer durch die Zeichen | getrennten Liste von Namenstokens, gefolgt von einer schliessenden Klammer
    - Ein Namenstoken ist ein Name, der aus einer beliebigen Zusammenstellung von Buchstaben, Zahlen, Punkten, Bindestrichen und Unterstrichen besteht (plus einem Doppelpunkt mit Namespace)

  Beispiel
    - <!ATTLIST FILM Class (fictional | instructional | documentray) „Fiction“>

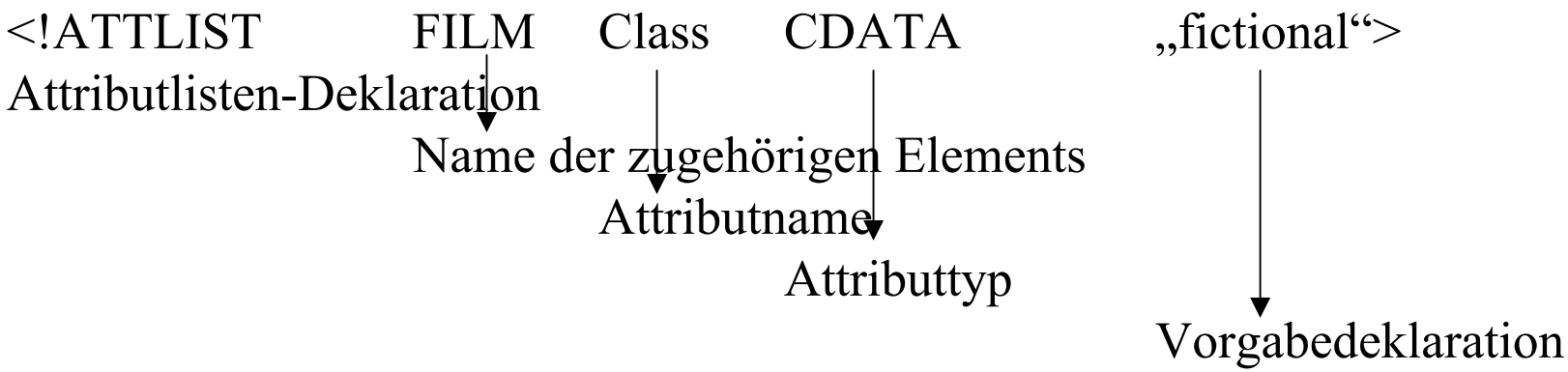
Gültige XML Dokumente Erstellen
Einen Aufzählungstyp Spezifizieren

```
<?xml version="1.0"?>
<!-- FilmMitAufzaehltyp.xml //-->
<!DOCTYPE FILM
    [
    <!ELEMENT FILM (TITLE, (STAR|NARRATOR|INSTRUCTOR))>
    <!ATTLIST FILM
            Class (fictional | instructional | documentary) "fictional">
    <!ELEMENT TITLE (#PCDATA)>
    <!ELEMENT STAR (#PCDATA)>
    <!ELEMENT NARRATOR (#PCDATA)>
    <!ELEMENT INSTRUCTOR (#PCDATA)>
    ]
>

<FILM Class="instructional"><!– falls Class fehlt, wird „fictional“ angenommen //-->
    <TITLE>The Use and Care of XML</TITLE>
    <NARRATOR>Michael Young</NARRATOR>
</FILM>
```

Gültige XML Dokumente Erstellen
Einen Aufzählungstyp Spezifizieren (2)

- Das Schlüsselwort NOTATION gefolgt von einem Leerzeichen, einer öffnenden Klammer, einer durch | getrennten Liste von Notationsnamen und einer schliessenden Klammer.
  Jeder dieser Namen muss mit einer in der DTD deklarierten Notation übereinstimmen.
  Eine Notation beschreibt ein Datenformat oder identifiziert das Programm, das ein bestimmtes Format verarbeitet.
  Beispiel:
  - <!ELEMENT BEISPIEL_DOC (#PCDATA)>
    <!ATTLIST BEISPIEL_DOC Format NOTATION (HTML | SGML | RTF) #REQUIRED>
  - <BEISPIEL_DOC Format=„HTML“><![CDATA[
    <HTML><HEAD><TITLE>Geht‘s noch</TITLE></HEAD>
    <BODY>Wie geht‘s?</BODY>
    </HTML>]]></BEISPIEL_DOC>

- Die Vorgabedeklaration ist die dritte und letzte erforderliche Komponente einer Attributdefinition. Sie legt fest, ob das Attribut erforderlich ist, und falls dies der Fall ist, zeigt sie an, wie der Prozessor reagieren soll, wenn das Attribut weggelassen wird. Beispiel:
  - 

- Die Vorgabedeklaration §kann von einem der folgenden vier Typen sein:
  - #REQUIRED: bei dieser Vorgabe müssen Sie für jedes Element des zugehörigen Typs einen Attributwert angeben (siehe Beispiel)
  - #IMPLIED: Diese Vorgabe zeigt an, dass Sie das Attribut eines Elements des zugehörigen Typs entweder aufnehmen oder weglassen können (optional ist).
    Beispiel:
    - <!ATTLIST FILM Class CDATA #IMPLIED>

Gültige XML Dokumente Erstellen
Die Vorgabedeklaration (3)

- *AttWert*: wobei AttWert für einen voreingestellten Attributwert steht. Sie können das Attribut bei einem Element entweder aufnehmen oder weglassen (dann verwendet der Prozessor den Vorgabewert).
  Beispiel:
  - <!ATTLIST FILM Class CDATA „fictional“>
  - <FILM>The Graduate</FILM> <! Ist gleichwertig mit... //--> <FILM Class=„fictional“>The Graduate</FILM>

Gültige XML Dokumente Erstellen
Die Vorgabedeklaration (4)

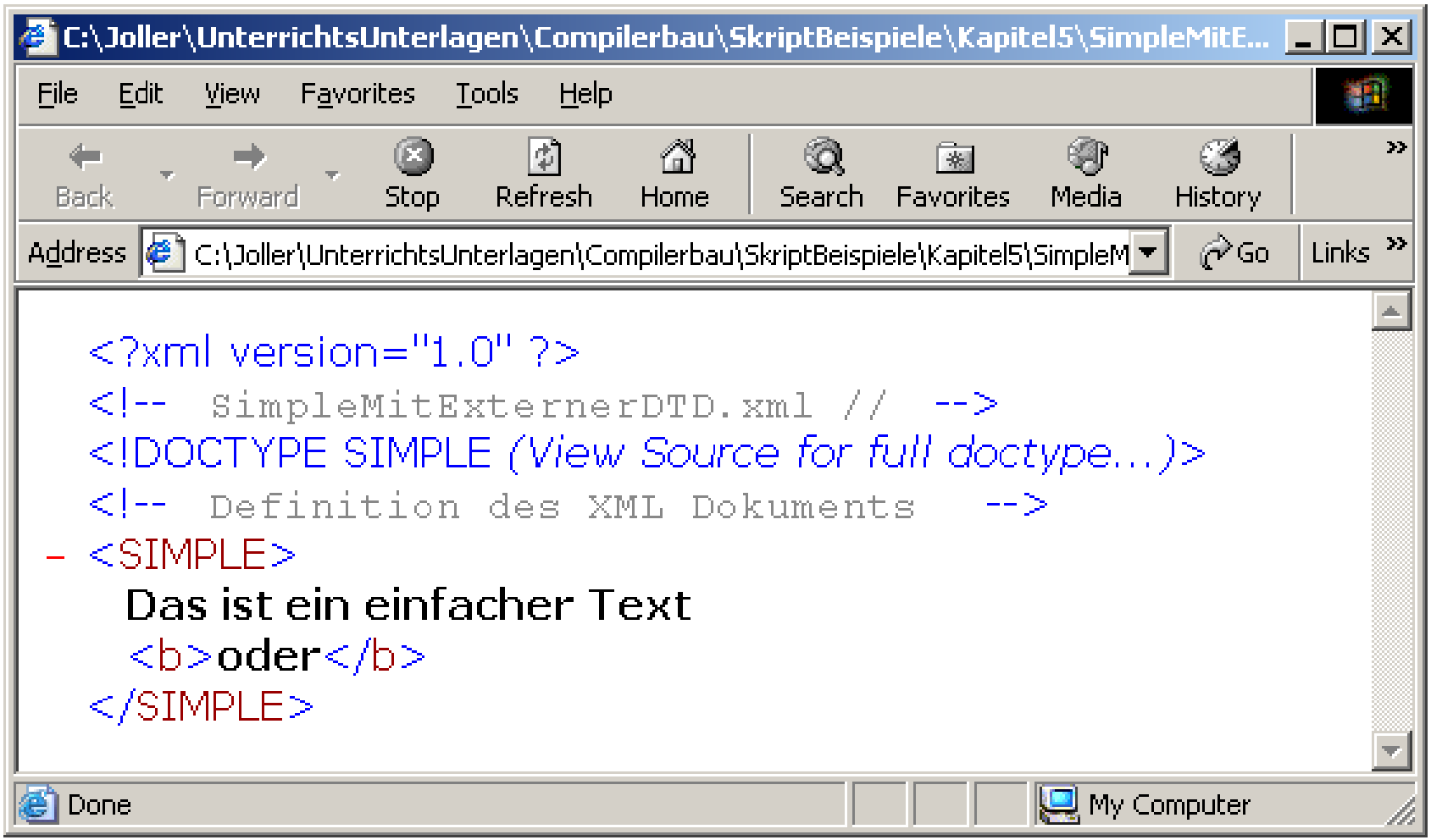
- #FIXED *AttrWert*: AttrWert steht dabei für einen voreingestellten Attributwert.
  Beispiel:
  - <!ATTLIST FILM Class CDATA #FIXED „documentary“>
  - <FILM>Herr der Ringe</FILM> ist äquivalent zu
    <FILM Class=„documentary“>Herr der Ringe</FILM>
    ungültig ist:
    <FILM Class=„instructional“>Herr der Ringe</FILM>

- Falls die DTD im Dokument selber steht, spricht man von *interner DTD Teilmenge*.
- Falls die DTD in einer separaten Datei steht, spricht man von einer *externen DTD Teilmenge.*
  - *Externe DTD Teilmengen* sind sehr praktisch, wenn man eine Dokumentstruktur unternehmensweit einführen möchte.

Gültige XML Dokumente Erstellen
Nur Eine Externe DTD Teilmenge Verwenden

- Bei externen DTD Teilmengen lassen Sie den Block in der eckigen Klammer weg und tragen statt dessen das Schlüsselwort `SYSTEM` gefolgt von der DTD Datei URL ein Beispiel:
  - ```
    <?xml version=„1.0“?>
    <!DOCTYPE SIMPLE
      [     <!ELEMENT SIMPLE ANY>   ]>
    <SIMPLE>Hier steht die DTD in der XML
    Datei</SIMPLE>
    ```
  - Simple.dtd: **`<!ELEMENT SIMPLE ANY>`**
    `(ohne <!DOCTYPE SIMPLE...)`
    ```
    XML: ...
         <!DOCTYPE SIMPLE SYSTEM „simple.dtd“>
         ...
    ```
    ```

Gültige XML Dokumente Erstellen
Nur Eine Externe DTD Teilmenge Verwenden



Gültige XML Dokumente Erstellen
Sowohl Interne Als Auch Externe DTD Teilmengen Verwenden

- Sie können interne mit externen DTD Teilmengen kombinieren.
  Beispiel:

```
<?xml version="1.0"?> <!-- Buch.xml //-->
<!DOCTYPE Buch SYSTEM "buchMixed.dtd"
    [
    <!ATTLIST Buch ISBN CDATA #IMPLIED Year CDATA "2002">
    <!ELEMENT Titel (#PCDATA)>
    ]>
<Buch Year="2001">
    <Titel>Das Universum in der Nussschale</Titel>
</Buch>

<!-- buchMixed.dtd //-->
    <!ELEMENT Buch ANY>
    <!ATTLIST Buch ISBN NMTOKEN #REQUIRED>
```

- Regeln:
  - Ist ein Attribut mehrfach vorhanden, wird die ERSTE Deklaration verwendet
  - Die interne DTD wird zuerst abgearbeitet

- Anwendung:
  - Sie können aus einer übergeordneten DTD spezielle DTDs ableiten, indem Sie interne DTDs als Ergänzung hinzufügen (diese überschreiben die allgemeine Definition)

- Sie können Teile einer DTD ausklammern oder einbeziehen, indem Sie den entsprechenden Abschnitt mit `<![IGNORE[ ...]]>` oder `<![INCLUDE[ ...]]>` ausklammern oder einbeziehen
  Beispiel:
  -
```
<![INCLUDE[
  <!- optionaler Block //-->
    <!ATTLIST BUCH Category CDATA „Fiktion“>
    <!ELEMENT TITEL (#PCDATA)>
    <!ELEMENT AUTOR (#PCDATA)>
]]>
```
  - Mit IGNORE würde der entsprechende Block ausgeklammert.

Gültige XML Dokumente Erstellen
Ein Wohlgeformtes Dokument in Ein Gültiges Umwandeln

1. Erstellen Sie das XML Dokument
2. Erstellen Sie eine passende DTD:
    1. fangen Sie mit <!DOCTYPE an
    2. Definieren Sie die Elemente
    3. Definieren Sie die Attribute

Gültige XML Dokumente Erstellen
Ein Wohlgeformtes Dokument in Ein Gültiges Umwandeln

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

<INVENTORY>
  <BOOK InStock="ja">
    <TITLE>The Adventures of Huckleberry Finn</TITLE>
    <AUTHOR Born="1835">Mark Twain</AUTHOR>
    <BINDING>Taschenbuch</BINDING>
    <PAGES>336</PAGES>
    <PRICE>DM 12,75</PRICE>
  </BOOK>
...
  <BOOK InStock="nein">
    <TITLE>Harry Potter und der Stein der Weisen</TITLE>
    <AUTHOR>Joanne K. Rowling</AUTHOR>
    <BINDING>Gebundene Ausgabe</BINDING>
    <PAGES>335</PAGES>
    <PRICE>DM 26,00</PRICE>
  </BOOK>
</INVENTORY>
```

Gültige XML Dokumente Erstellen
Ein Wohlgeformtes Dokument in Ein Gültiges Umwandeln

```
<!DOCTYPE INVENTORY
  [
  <!ELEMENT INVENTORY (BOOK)*>
  <!ELEMENT BOOK (TITLE, AUTHOR, BINDING, PAGES, PRICE)>
  <!ATTLIST BOOK   InStock (ja|nein) #REQUIRED>

  <!ELEMENT TITLE (#PCDATA | SUBTITLE)*>

  <!ELEMENT SUBTITLE (#PCDATA)>

  <!ELEMENT AUTHOR (#PCDATA)>
  <!ATTLIST AUTHOR   Born CDATA #IMPLIED>

  <!ELEMENT BINDING (#PCDATA)>

  <!ELEMENT PAGES (#PCDATA)>

  <!ELEMENT PRICE (#PCDATA)>
  ]
>
```

Gültige XML Dokumente Erstellen
Ein Wohlgeformtes Dokument in Ein Gültiges Umwandeln