

XML – Eine Einführung

XML-Dokumente Im Internet Darstellen

XML – Eine Einführung

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von DOM-
Skripts Anzeigen

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von DOM-Skripts Anzeigen

- Sie lernen in diesem Kapitel
 - Wie man XML-Dokumente mit HTML-Dokumenten verknüpfen kann
 - Wie man unter Verwendung des DOM auf das Dokument zugreifen kann
 - Die Struktur von DOM kennen
 - Methoden
 - Techniken für den Zugriff auf Attribute, Entities und Notationen

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von DOM-Skripts Anzeigen

- Die im folgenden beschriebene Technik wurde von W3C entwickelt und als Standard publiziert.
 - Link
 - <http://www.w3c.org/TR/REC-DOM-Level-1>
 - DOM steht für *Document Object Model*
 - Damit können beliebige XML Dokumente angezeigt und bearbeitet werden.
 - Sie können auf die Dokumente-Komponenten mittels im API definierten Methoden zugreifen und die Komponenten verändern.
 - Das API steht für verschiedene Programmiersprachen (z.B. Java) und Skriptsprachen (z.B. JavaScript) zur Verfügung.

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von DOM-Skripts Anzeigen

XML-Dokument Mit Einer HTML-Seite Verknüpfen

- Die Verknüpfung eines XML-Dokuments mit einer HTML-Seite ist gemäss DHTML auf unterschiedliche Art und Weise möglich
 - Microsoft verwendet dazu den `<XML>` Tag in HTML
 - Der Internet Explorer besitzt einen eingebauten Parser, der in der Lage ist, XML-Dokumente zu analysieren und intern als Baum zu speichern
 - `<HTML>...`
`<BODY>`
`<XML ID=„domBOOK“ SRC=„Book.xml“></XML> ...`
`</HTML>`
 - In der Regel verwendet man externe Parser (z.B. Xerxes von Apache)

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von DOM-Skripts Anzeigen

DOM Baum

```
- <?xml version="1.0"?>
  <!-- Dateiname: Inventory Dom.xml -->
  <INVENTORY>
    <BOOK Binding="Taschenbuch">
      <TITLE>The Adventures of Huckleberry Finn</TITLE>
      <AUTHOR Born="1835">Mark Twain</AUTHOR>
      <PAGES>336</PAGES>
      <PRICE>EUR 12,75</PRICE>
    </BOOK>
    <BOOK Binding="Taschenbuch">
      <TITLE>Der Graf von Monte Christo</TITLE>
      <AUTHOR Born="1802">Alexandre Dumas</AUTHOR>
      <PAGES>760</PAGES>
      <PRICE>EUR 38,00</PRICE>
    </BOOK>
    <BOOK Binding="Gebundene Ausgabe">
      <TITLE>Moby-Dick</TITLE>
      <AUTHOR Born="1819">Herman Melville</AUTHOR>
      <PAGES>724</PAGES>
      <PRICE>EUR 44,00</PRICE>
    </BOOK>
  </INVENTORY>
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von DOM-Skripts Anzeigen

DOM Baum

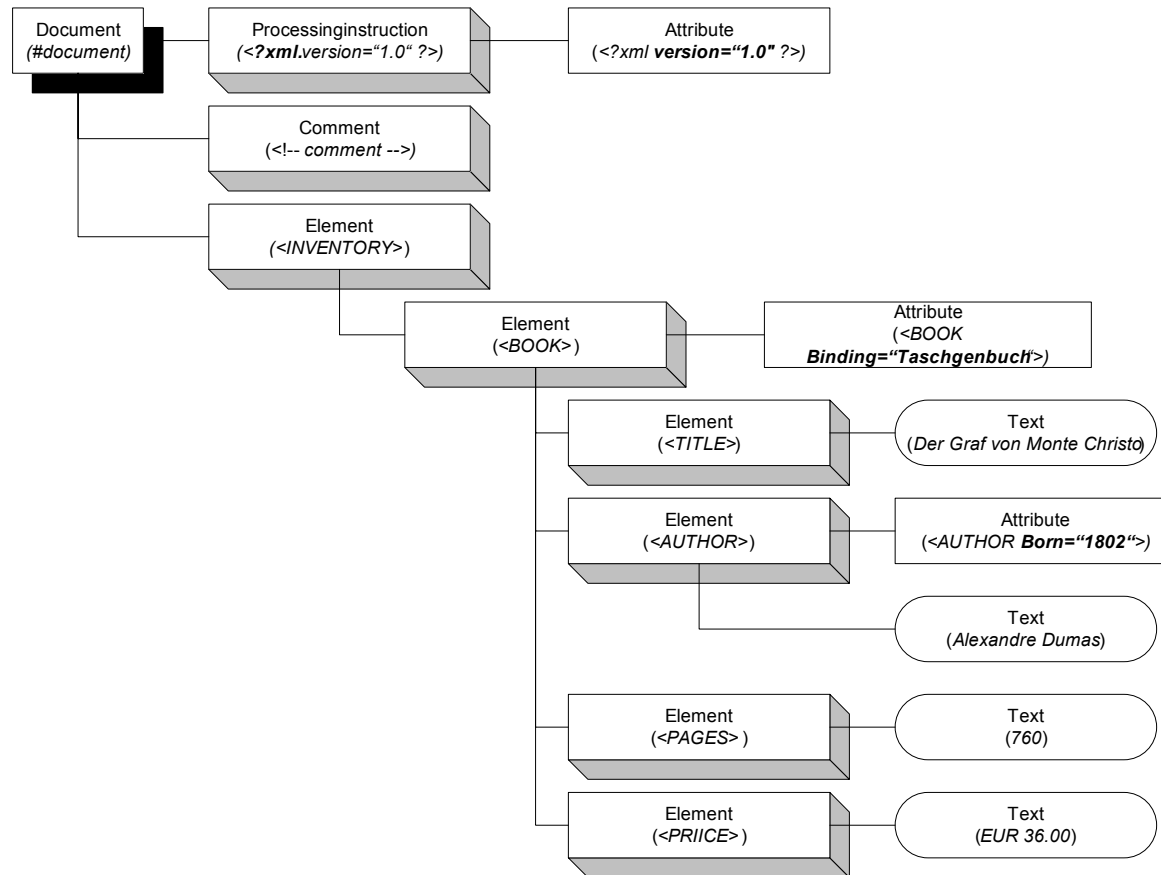
– Als Hierarchie sieht dieses Dokument folgendermassen aus:

```
- Document (#document)
  ProcessingInstruction (xml)
    Attribute (version)
  Comment (#comment)
  Element (INVENTORY)
    Element (BOOK)
      Attribute (Binding)
      Element (TITLE)
        Text (#text)
      Element (AUTOR)
        Attribute (Born)
        Text (@text)
      Element (PAGES)
        Text (#text)
      Element (PRICE)
        Text (@text)
    Element (BOOK)
      Attribute (Binding)
      Element (TITLE)
        Text (#text)
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von DOM-Skripts Anzeigen

DOM Baum



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Struktur Des Dokumentenobjektmodell (DOM)

- DOM repräsentiert die Programmierobjekte der *XML-Dokumente* als *Knoten*
- Unterschiedliche *XML-Komponenten* werden durch unterschiedliche *Knotentypen* repräsentiert
 - Wurzelknoten / gesamtes Dokument *Document*
 - XML-Element *Element*
 - Text eines Elements, Attributs, Entity *Text*
 - XML-Attribut bzw. generell (Name, Wert) *Attribute*
 - XML-Verarbeitungsanweisung *ProcessingInstruction*
 - XML-Kommentar *Comment*
 - CDATA-Abschnitt *CDATASection*
 - Dokumententypdeklaration *DocumentType*
 - DTD-Entity *Entity*
 - DTD-Notationsdeklaration *Notation*
- Wobei Sie jeweils *Knotenname* (Objekteigenschaft *nodeName*) und *Knotenwert* (Objekteigenschaft *nodeValue*) abfragen können.

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Struktur Des Dokumentenobjektmodell (DOM)

- Zusätzlich stehen für *jeden* Knotentyp folgende Eigenschaften zur Verfügung
 - **attributes** *NamedNodeMap*-Auflistung
 - *AttributeNode = Element.attributes.getNamedItem(„Binding“);*
 - **childNodes** *NodeList* Auflistung untergeordneter Knoten
 - *FirstNode = Element.childNodes(0);*
 - **dataType** Datentyp des Knotens
 - *AttributeType = Attribute.dataType;*
 - **firstChild** erster untergeordneter Knoten
 - *FirstChildNode = Element.firstChild;*
 - **lastChild** letzter untergeordneter Knoten
 - **nextSibling** auf gleicher Hierarchiestufe nachfolgender Knoten
 - **nodeName** Name des Knotens
 - **nodeType** numerischer Code (Kontentyp)

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Struktur Des Dokumentenobjektmodell (DOM)

- Zusätzlich stehen für *jeden* Knotentyp folgende Eigenschaften zur Verfügung (Fortsetzung)
 - **nodeTypeString** Type des Knotens als Zeichenkette
 - **nodeValue** Wert des Knotens oder *null*
 - **ownerDocument** Wurzelknoten dieses Dokuments
 - **parentNode** Knoten, dem dieser untergeordnet ist
 - **previousSibling** vorheriger Knoten (selbe Hierarchiestufe)
 - **text** Textinhalt dieses Knotens
 - **xml** XML-Inhalt dieses Knotens und aller untergeordneten *Element*-Knoten
 - Eine vollständige Liste aller Knotenobjekte, Eigenschaften, Methoden und Ereignisse finden Sie in der Referenz zum jeweiligen Browser bzw. bei W3C
 - Eine Übersicht finden Sie in selfHTML

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Praktisches Beispiel – Übertragen Des Textes Aus XML Ins HTML

- Im folgenden *Beispiel* werden die Texte aus den Elementbeschreibungen eines XML-Dokuments mittels JavaScript in ein HTML-Dokument übertragen

– XML

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

```
<!-- Dateiname: Book.xml -->
```

```
<BOOK>
```

```
  <TITLE>The Adventures of Huckleberry Finn</TITLE>
```

```
  <AUTHOR>Mark Twain</AUTHOR>
```

```
  <BINDING>Taschenbuch</BINDING>
```

```
  <PAGES>336</PAGES>
```

```
  <PRICE>EUR 12,75</PRICE>
```

```
</BOOK>
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Praktisches Beispiel – Übertragen Des Textes Aus XML Ins HTML

– HTML-Fragment

- `<BODY>`
 - `<XML ID="dsoBook" SRC="Book.xml"></XML>`
 - `<H2>Buchbeschreibung</H2>`
 - `Titel: `
 - ``
 - `
`
 - `Autor: `
 - ``
 - `
`
 - `Bindung: `
 - ``
 - `
`
 - `Seitenzahl: `
 - ``
 - `
`
 - `Preis: `
 - ``

`</BODY>`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Praktisches Beispiel – Übertragen Des Textes Aus XML Ins HTML

– HTML-Fragment

- Zur Erinnerung (aus selfHTML)
- <SPAN...>
 - “Analog zum [div](#)-Element, das andere Block-Elemente enthalten kann, gibt es ein Element, das Text und andere Inline-Elemente enthalten kann, selbst aber keinerlei Eigenschaften hat und nichts bewirkt. Es ist dazu gedacht, um mit Hilfe von **CSS formatiert** zu werden”
- Konkret
 - Wir können das XML-Dokument mit der HTML-Seite verknüpfen (wie bereits gehabt)
 - Dann können wir über die ID auf den Inhalt des Elements zugreifen bzw. Text setzen (aus XML übernehmen)

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Praktisches Beispiel – Übertragen Des Textes Aus XML Ins HTML

– JavaScript

- `<SCRIPT LANGUAGE="JavaScript" FOR="window,, EVENT="ONLOAD">`
 `Document = dsoBook.XMLDocument;`
 title.innerText=
 `Document.documentElement.childNodes(0).text;`
 author.innerText=
 `Document.documentElement.childNodes(1).text;`
 binding.innerText=
 `Document.documentElement.childNodes(2).text;`
 pages.innerText=
 `Document.documentElement.childNodes(3).text;`
 price.innerText=
 `Document.documentElement.childNodes(4).text;`
`</SCRIPT>`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Praktisches Beispiel – Übertragen Des Textes Aus XML Ins HTML

– JavaScript Interpretation

- `<SCRIPT LANGUAGE="JavaScript" FOR="window" EVENT="ONLOAD">`
 - *Beim Laden des Fensters wird der Skriptcode ausgeführt*
- `Document = dsoBook.XMLDocument;`
 - *Bestimmen der Dokumentwurzel*
- `title.innerText=Document.documentElement.childNodes(0).text;`
 - *Bestimme den Text aus dem ersten ChildNode und ordne diesen dem HTML SPAN Element mit der **ID=title** zu*
- `author.innerText=Document.documentElement.childNodes(1).text;`
 - *Analog zur obigen Zeile*

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Praktisches Beispiel – Übertragen Des Textes Aus XML Ins HTML

– Anzeige

- Dateien
 - DOM Demo Fixed.htm
 - Book.xml



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Des *Document*-Knotens

- **doctype** *DocumentType*-Knoten
(Dokumenttypdeklaration)
 - `DocumentType=Document.doctype`
- **documentElement** Wurzelement *Element*-Knoten
 - `RootElement=Document.documentElement;`
- **ondataavailable** zugeordnete Funktion wird aufgerufen, sobald XML-Daten verfügbar sind
 - `Document.ondataavailable=MeinDataAvailableHandler;`
- **onreadystatechange** Handler der *readyState* Eigenschaft
- **parseError** Informationen zu Fehlern
 - `ErrorCode=Document.parseError.errorCode;`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Des *Document*-Knotens

- **readyState** Lade-/Verarbeitungszustand
 - 0=nicht initialisiert 1=wird geladen 2=ist geladen
 - 3=interaktiv 4=abgeschlossen
 - If (Document.readyState == 4) ...
- **url** URL des XML-Dokuments
 - URL=Document.url;

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Methoden Des *Document*-Knotens

- Mithilfe der *Methoden* lassen sich Eigenschaften der XML-Dokuments (*Document*) abfragen
 - **getElementsByTagName(*Typname*)**
 - Liefert eine NodeList-Auflistung der im Dokument enthaltenen Elemente des angegebenen Typs
 - **AuthorElementCollection =**
Document.getElementsByTagName(„AUTHOR“);
 - **nodeFromID(*ID-Wert*)**
 - Liefert den Knoten zum Element mit der angegebenen ID
 - **Element=Document.nodeFromID(„T1022“);**
wobei im HTML-Dokument (hoffentlich) diese ID vorkommt

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Und Methoden Des *NodeList*-Objekts

- Listen der untergeordneten Knoten werden beispielsweise durch die *childNodes* Eigenschaft geliefert.
 - Diese enthält eine bestimmte Anzahl Knoten
 - ***NodeList*-Eigenschaften**
 - **length**
 - gibt die Anzahl der in der Liste enthaltenen Knoten
 - Beispiel
 - » **NodeCount = Element.childNodes.length;**

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Und Methoden Des *NodeList*-Objekts

– ***NodeList*-Methoden**

- **item(*Index*)** (Standard)
 - Liefert den Knoten, der sich an der *Index*-ten Position +1 befindet (der *Index* startet mit 0)
 - Beispiel
 - » **SecondChild = Element.childNodes.item(1);**
- **reset()**
 - Setzt den Zeiger vor den ersten Knoten
 - » **Element.childNodes.reset();**
- **nextNode()**
 - Liefert den nächsten Knoten
 - » **Element.childNodes.reset();**
FirstName = Element.childNodes.nextNode();

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Und Methoden Des *Text*-Knotens

- Wie können wir auf den Text der Elemente zugreifen?
 - Beispiel
 - `Title.InnerText =`
`document.documentElement.childNodes(0).text;`
 - Dieses Verfahren funktioniert nur, falls keine untergeordneten Elemente vorhanden sind
 - Andernfalls muss auf den DOM *Text*-Knoten zugegriffen werden
 - Dafür stehen Methoden zur Verfügung
der einem Element zugeordnete *Text*-Knoten besitzt auch bestimmte Eigenschaften,
für Textinformationen typische Eigenschaften

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Und Methoden Des *Text*-Knotens

– Eigenschaften

- **length**

- Die Zeichenanzahl des Textinhalts des Knotens

- Beispiel

- » `CharacterCount = Text.length;`

– Methoden

- **substringData(*Zeichenposition*, *ZeichenAnzahl*)**

- Liefert die Teilzeichenkette mit der Anzahl *ZeichenAnzahl* Zeichen, ab der Zeichenketten-Position *Zeichenposition*, wobei mit der Position 0 gestartet wird

- Beispiel

- » `SubString = Text.substringData(2,3);`
`/* ab Position 2; Länge = 3 Zeichen */`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Und Methoden Des *Text-Knotens*

– Beispiel

- Zugriff auf Text ohne Verschachtelung

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

- `<BUCH>`

- `<TITEL>Java als erste Programmiersprache </TITEL>`

- `<UNTERTITEL>Eine Einfuehrung</UNTERTITEL>`

- `<AUTHOR>Joachim Goll</AUTHOR>`

- `<BUCHTYP>Gebunden</BUCHTYP>`

- `<SEITEN>880</SEITEN>`

- `</BUCH>`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

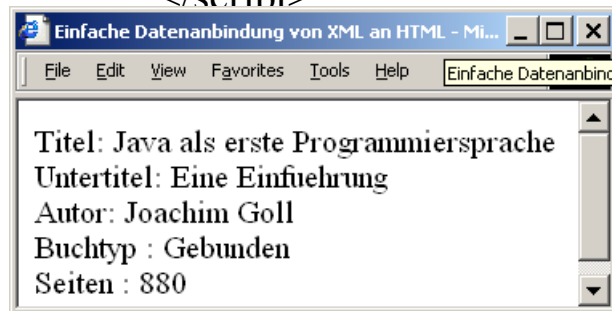
Eigenschaften Und Methoden Des *Text*-Knotens

– Beispiel

- Zugriff auf Text ohne Verschachtelung

- JavaScript

```
» <script language="JavaScript" for="window" event="ONLOAD">  
Document=domBook.XMLDocument;  
titel.innerText=Document.documentElement.childNodes(0).text;  
untertitel.innerText=Document.documentElement.childNodes(1).text;  
author.innerText=Document.documentElement.childNodes(2).text;  
buchtyp.innerText=Document.documentElement.childNodes(3).text;  
seiten.innerText=Document.documentElement.childNodes(4).text;  
</script>
```



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Und Methoden Des *Text*-Knotens

– Beispiel

- Zugriff auf Text mit Verschachtelung

```
– <?xml version="1.0" ?>
  <BUCH>
    <TITEL>Java als erste Programmiersprache
    <UNTERTITEL>Eine Einfuehrung</UNTERTITEL>
  </TITEL>
  <AUTHOR>Joachim Goll</AUTHOR>
  <BUCHTYP>Gebunden</BUCHTYP>
  <SEITEN>880</SEITEN>
</BUCH>
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

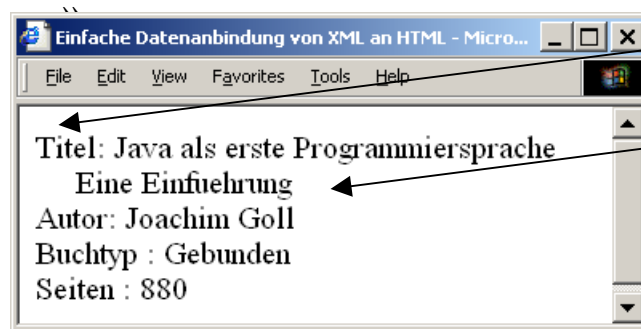
Eigenschaften Und Methoden Des *Text*-Knotens

– Beispiel

- Zugriff auf Text *mit* Verschachtelung

- JavaScript

```
» <script language="JavaScript" for="window" event="ONLOAD">
  Document=domBook.XMLDocument;
  titel.innerText=Document.documentElement.childNodes(0).text;
  author.innerText=Document.documentElement.childNodes(1).text;
  buchtyp.innerText=Document.documentElement.childNodes(2).text;
  seiten.innerText=Document.documentElement.childNodes(3).text;
</script>
```



in diesem Fall wird der Text des Titels und des Untertitels gemeinsam angezeigt.

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Und Methoden Des *Text-Knotens*

- Ein weiteres Beispiel

- Eine *variable* Anzahl XML-Elemente Anzeigen wobei die Anzahl mithilfe der **length** Eigenschaft bestimmt wird

- XML

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

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

- `<INVENTORY>`

- `<BOOK>`

- `<TITLE>The Adventures of Huckleberry Finn</TITLE>`

- `<AUTHOR>Mark Twain</AUTHOR>`

- `<BINDING>Taschenbuch</BINDING>`

- `<PAGES>336</PAGES>`

- `<PRICE>EUR 12,75</PRICE>`

- `</BOOK>`

- `<BOOK> ... </BOOK>`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

Eigenschaften Und Methoden Des *Text*-Knotens

- Ein weiteres Beispiel

- Eine *variable* Anzahl XML-Elemente Anzeigen wobei die Anzahl mithilfe der **length** Eigenschaft bestimmt wird

- JavaScript

- ```
<SCRIPT LANGUAGE="JavaScript" FOR="window" EVENT="ONLOAD">
HTMLCode = "";
Document = dsoInventory.XMLDocument;
for (i=0; i < Document.documentElement.childNodes.length; i++) {
 HTMLCode += "Titel: “
+ Document.documentElement.childNodes(i).childNodes(0).text
+ "
“
+ "Autor: “
+ Document.documentElement.childNodes(i).childNodes(1).text
+ "
“
+ "Bindung: “
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Eigenschaften Und Methoden Des *Text*-Knotens**

- Ein weiteres Beispiel

- Eine *variable* Anzahl XML-Elemente Anzeigen wobei die Anzahl mithilfe der **length** Eigenschaft bestimmt wird

- JavaScript

```
+ Document.documentElement.childNodes(i).childNodes(3).text
+ "
"
+ "Preis: "
+ Document.documentElement.childNodes(i).childNodes(4).text
 + "<P>";
 }
 DisplayDIV.innerHTML=HTMLCode;
</SCRIPT>
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Eigenschaften Und Methoden Des *Text*-Knotens**

- Ein weiteres Beispiel

- Eine *variable* Anzahl XML-Elemente Anzeigen wobei die Anzahl mithilfe der **length** Eigenschaft bestimmt wird

- HTML

- <BODY>

- <XML ID="dsoInventory" SRC="Inventory.xml"></XML>

- <H2>Buchbestand</H2>

- <DIV ID="DisplayDIV"></DIV>

- </BODY>



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Eigenschaften Und Methoden Des *Text*-Knotens**

- Zur Erinnerung

- `<DIV></DIV>`

- Nach selfHTML

- “Sie können mehrere Elemente wie Text, Grafiken, Tabellen usw., in einen gemeinsamen Bereich einschließen. Dieses allgemeine Element bewirkt nichts weiter als dass es in einer neuen Zeile des Fließtextes beginnt. Ansonsten hat es keine Eigenschaften. Es ist dazu gedacht, um mit Hilfe von CSS formatiert zu werden.”

- Konkret

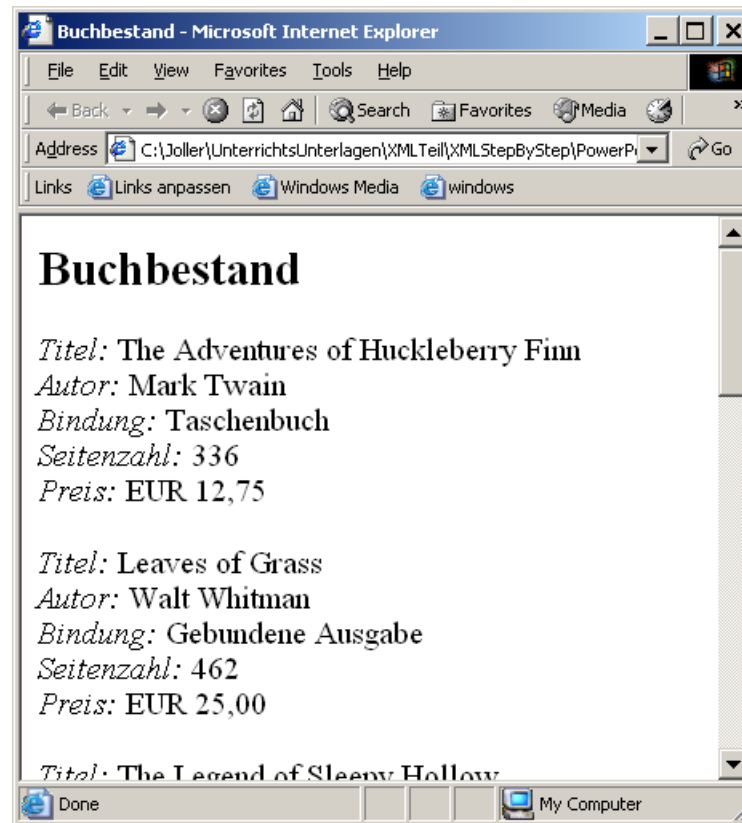
- » Wir können mit dem `<DIV>` Tag dynamisch erstellten HTML Code darstellen.

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Eigenschaften Und Methoden Des *Text*-Knotens

– Anzeige



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Eigenschaften Und Methoden Des *Element-Knotens*

– Was wir bereits gezeigt haben

- Sie können auf Element-Knoten zugreifen, indem Sie die Knotenhierarchie Knoten für Knoten durchlaufen.
- Methoden / Eigenschaften

– *childNodes*

– *firstChild*

Oder

– *lastChild*

– *previousSibling*

– *nextSibling*

– *parentNode*

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Eigenschaften Und Methoden Des *Element-Knotens*

### – Nachteil dieser Methoden / Eigenschaften

- Sie können mit den Eigenschaften *childNodes*, *firstNodes* und *lastChild* lediglich auf untergeordnete Knoten zugreifen, die **nicht** vom Knotentyp *Attribute* sind
- Sie können mit den Eigenschaften *previousSibling*, *nextSibling* auf beliebige Nachbarknoten der **gleichen** Hierarchieebene zugreifen.

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Eigenschaften Und Methoden Des *Element-Knotens*

### – Nachteil dieser Methoden / Eigenschaften

- Sie können mit den Eigenschaften *childNodes*, *firstNodes* und *lastChild* lediglich auf untergeordnete Knoten zugreifen, die **nicht** vom Knotentyp *Attribute* sind
- Sie können mit den Eigenschaften *previousSibling*, *nextSibling* auf beliebige Nachbarknoten der **gleichen** Hierarchieebene zugreifen.

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Eigenschaften Und Methoden Des *Element-Knotens*

- Andere (weitere) Möglichkeiten / Methoden
  - **getAttribute**
    - Liefert den Wert des angegebenen Element-Attributs
    - Beispiel
      - » `AttValue = Element.getAttribute(„InStock“);`
  - **getAttributeNode**
    - Liefert den Attribute-Knoten zum gegebenen Element-Attribut
    - Beispiel
      - » `Attribute = Element.getAttributeNode(„InStock“);`
  - **getElementsByTagName(*Typname*)**
    - Liefert ein NodeList-Auflistungsobjekt mit den Element-Knoten aller untergeordneten Elemente dieses Elements mit passendem Typnamen bzw.alle falls „\*“ als Parameter angegeben wird.
    - Beispiel
      - » `AuthorElementCollection = Element.getElementsByTagName(„AUTHOR“);`

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Eigenschaften Und Methoden Des *Element-Knotens*

#### – Anwendungsbeispiel

- Elemente über Elementnamen suchen

##### – JavaScript

```
» function ShowElements() {
 /* Sicherstellen, dass der Benutzer Text in das Feld 'Elementname' eingegeben hat: */
 if (ElementName.value == "") {
 ResultDiv.innerHTML = "<Sie müssen einen Elementnamen "
 + "in das Feld 'Elementname' eingeben.>";
 return; }
 /* Eine NodeList-Auflistung aller passenden Element-Knoten des Dokuments */
 Document = domXML.XMLDocument;
 NodeList = Document.getElementsByTagName (ElementName.value);
 /* Die XML-Auszeichnung für jeden Element-Knoten in ResultHTML speichern: */
 ResultHTML = "";
 for (i=0; i < NodeList.length; ++i)
 ResultHTML += NodeList(i).xml + "\n\n";
 /* Die gespeicherten Ergebnisse der Eigenschaft innerText von DIV zuweisen: */
 if (ResultHTML == "")
 ResultDiv.innerHTML = "<keine passenden Elemente gefunden>";
 else
 ResultDiv.innerHTML = ResultHTML;
 }
}
```

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Eigenschaften Und Methoden Des *Element-Knotens*

#### – Anwendungsbeispiel

- Elemente über Elementnamen suchen

##### – HTML

» <BODY>

<!-- Verknüpfung mit dem XML-Dokument //-->

<XML ID="domXML" SRC="Inventory.xml"></XML>

<H2>Elemente über Elementnamen suchen</H2>

<!-- Eingabe des Suchbegriffs //-->

Elementname: <INPUT TYPE="TEXT" ID="ElementName">&nbsp;

<BUTTON ONCLICK="ShowElements()">Elemente anzeigen</BUTTON>

<HR>

<!-- Anzeige des Ergebnisses //-->

<DIV ID=ResultDiv></DIV>

</BODY>

</HTML>



## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

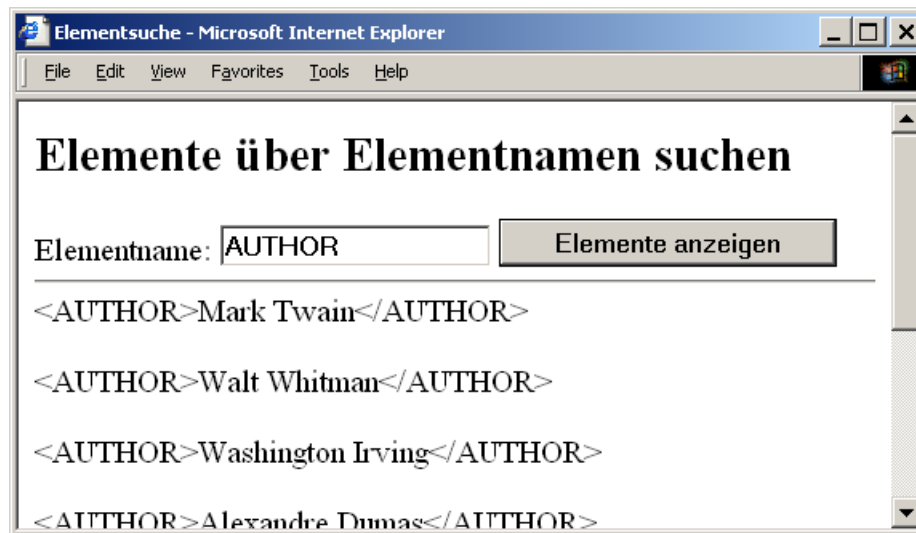
## Eigenschaften Und Methoden Des *Element-Knotens*

### – Anwendungsbeispiel

- Elemente über Elementnamen suchen

#### – Anzeige

» DOM GetElements.htm und Inventory.xml



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Eigenschaften Und Methoden Des *NamedNodeMap***

– Motivation

- DOM repräsentiert Attribute-Knoten und verschiedene andere Typen von XML-Komponenten durch Name-Wert-Paare
  - Name und Wert der Version in der XML-Deklaration
    - » *version=„1.0“*
  - Schlüsselwort SYSTEM und Systemliteral (DTD, externe Entities oder Notationsdeklaration)
  - NDATA und Notationsnamen in der Deklaration eines nicht geparsten Entity

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Eigenschaften Und Methoden Des *NamedNodeMap*

- Motivation

- XML Beispiel

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

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

- `<INVENTORY>`

- `<BOOK Binding="Taschenbuch" InStock="ja" Review="***">`

- `<TITLE>The Adventures of Huckleberry Finn</TITLE>`

- `<AUTHOR Born="1835">Mark Twain</AUTHOR>`

- `<PAGES>336</PAGES>`

- `<PRICE>EUR 12,75</PRICE>`

- `</BOOK>`

- `<BOOK Binding="Gebundene Ausgabe" InStock="nein">`

- `<TITLE>Leaves of Grass</TITLE>`

- `<AUTHOR Born="1819">Walt Whitman</AUTHOR>`

- `<PAGES>462</PAGES>`

- `<PRICE>EUR 25,00</PRICE>`

- `</BOOK>`

- `<BOOK Binding="Taschenbuch" InStock="ja,,> ...`

- `</BOOK>`

- `</INVENTORY>`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Eigenschaften Und Methoden Des *NamedNodeMap***

– Motivation

- Bestimmen der Attribute des BOOK-Elements
  - Erster Schritt: bestimmen des BOOK-Knotens
    - » `bookNode=Document.documentElement.childNodes(0)`
  - Zweiter Schritt: bestimmen der attributes-Eigenschaft
    - » `NamedNodeMap = bookNode.attributes;`

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Eigenschaften Und Methoden Des *NamedNodeMap*

### – Eigenschaft

- **getAttribute**

- Liefert den Wert des angegebenen Element-Attributs

- Beispiel

- » `AttValue = Element.getAttribute(„InStock“);`

- **getAttributeNode**

- Liefert den Attribute-Knoten zum gegebenen Element-Attribut

- Beispiel

- » `Attribute = Element.getAttributeNode(„InStock“);`

- **getElementsByTagName(*Typname*)**

- Liefert ein NodeList-Auflistungsobjekt mit den Element-Knoten aller untergeordneten Elemente dieses Elements mit passendem Typnamen bzw.alle falls „\*“ als Parameter angegeben wird.

- Beispiel

- » `AuthorElementCollection =  
Element.getElementsByTagName(„AUTHOR“);`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Eigenschaften Und Methoden Des *NamedNodeMap***

– **Eigenschaft**

• **length**

– Enthält die Anzahl Knoten im *NamedNodeMap*

– Beispiel

» **AttributeCount = Element.attributes.length;**

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Eigenschaften Und Methoden Des *NamedNodeMap*

#### – Methoden

- **getNamedItem(*Attributname*)**
  - Liefert den Knoten, der den Namen *Attributname* besitzt
  - Beispiel
    - » `Attribute = Element.attributes.getNamedItem(„BINDING“)`
- **item(*indexposition*)**
  - Liefert den Knoten an Position *indexposition*
    - » `SecondAttribute = Element.attributes.item(1);`
    - » `SecondAttribute = Element.attributes (1);`
- **reset()**
  - Interner Zeiger vor den ersten Knoten setzen
  - Beispiel
    - » `Element.attributes.reset();`
- **nextNode()**
  - Liefert den nächsten Knoten der Auflistung
  - Beispiel
    - » `Element.attributes.reset();`
    - » `FirstAttribute = Element.attributes.nextNode();`

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Eigenschaften Und Methoden Des *NamedNodeMap*

#### – Anwendungsbeispiele

- *length*-Eigenschaft und *NamedNodeMap* (*nodeValue* , *nodeName*)
  - Wird benötigt, um die *NamedNodeMap*-Objekte zu durchsuchen
  - Beispiel: Anzeige des ersten (Name, Wert) Attribute-Paar

» **NamedNodeMap =**

**Document.documentElement.childNodes(0).attributes;**

**for (i=0; i<NamedNodeMap.length; i++)**

**alert(„Name: „+NamedNodeMap(1).nodeName+“\n“  
+“Wert: „+NamedNodeMap(1).nodeValue);**

» **<BOOK Binding="Taschenbuch" InStock="ja" Review="\*\*\*">**

» Datei : Dom NamedNodeMap.htm



XML



(C) J.M.Joller





## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Eigenschaften Und Methoden Des *NamedNodeMap*

### – Anwendungsbeispiele

- *getNamedItem(Attributname)*-Methode

– Beispiel: Gezielte Anzeige der Werte zu einem bestimmten Attribut

» **NamedNodeMap =**

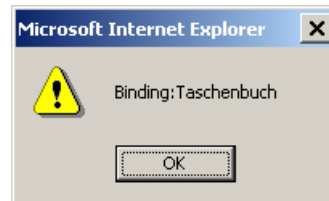
**Document.documentElement.childNodes(0).attributes;**

**alert(„Binding:„,**

**+NamedNodeMap.getNamedItem(„BINDING“).nodeValue;)**

» **<BOOK Binding="Taschenbuch" InStock="ja" Review="\*\*\*">**

» Datei : DOM NamedNodeMap NamedItem.htm



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### **Bemerkung Zu *Attribute* Knoten**

- Attribute in einem XML-Dokument werden durch einen untergeordneten *Attribute*-Node repräsentiert
- Sie können aber nicht auf die Eigenschaften mithilfe der Knoteneigenschaft *childNodes*, *firstChild* oder *lastChild* zugreifen
- Sie müssen dazu die ***attributes*-Eigenschaft** des *Element*-Knoten verwenden.

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Anwendungsbeispiel: Kombination Von *Attribute* Und *attributes* Und DTD**

- Ziel

- Wir wollen die Attributwerte eines Attribute-Knoten analysieren

- Motivation

- Die Syntax eines XML-Dokuments sei durch eine DTD beschrieben.

- Diese DTD enthält ENTITY und ATTLIST Elemente

- Wir möchten die Werte der im XML-Dokument angegebenen Attribut (also der ENTITY und ATTLIST) bestimmen

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Anwendungsbeispiel: Kombination Von *Attribute* Und *attributes* Und DTD

- DTD

```
– <!DOCTYPE INVENTORY
 [
 <!NOTATION TXT SYSTEM "Nur Text (txt)">
 <!ENTITY rev_huck SYSTEM "Review Huckleberry Finn.txt"
 NDATA TXT>
 <!ENTITY rev_leaves SYSTEM "Review Leaves of Grass.txt"
 NDATA TXT>
 <!ENTITY rev_legend SYSTEM "Review Sleepy Hollow.txt"
 NDATA TXT>

 <!ELEMENT INVENTORY (BOOK)*>
 <!ELEMENT BOOK (TITLE, AUTHOR, BINDING, PAGES, PRICE)>
 <!ATTLIST BOOK Review ENTITY #IMPLIED>
 <!ELEMENT TITLE (#PCDATA)>
 <!ELEMENT AUTHOR (#PCDATA)>
 <!ELEMENT BINDING (#PCDATA)>
 <!ELEMENT PAGES (#PCDATA)>
 <!ELEMENT PRICE (#PCDATA)>
]
>
```

## XML-Dokumente Im Internet Darstellen

### XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Anwendungsbeispiel: Kombination Von *Attribute* Und *attributes* Und DTD

- XML

- <INVENTORY>

- <BOOK Review="rev\_huck">

- <TITLE>The Adventures of Huckleberry Finn</TITLE>

- <AUTHOR>Mark Twain</AUTHOR>

- <BINDING>Taschenbuch</BINDING>

- <PAGES>336</PAGES>

- <PRICE>EUR 12,75</PRICE>

- </BOOK>

- <BOOK Review="rev\_leaves">

- <TITLE>Leaves of Grass</TITLE>

- <AUTHOR>Walt Whitman</AUTHOR>

- <BINDING>Gebundene Ausgabe</BINDING>

...

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Anwendungsbeispiel: Kombination Von *Attribute* Und *attributes* Und DTD

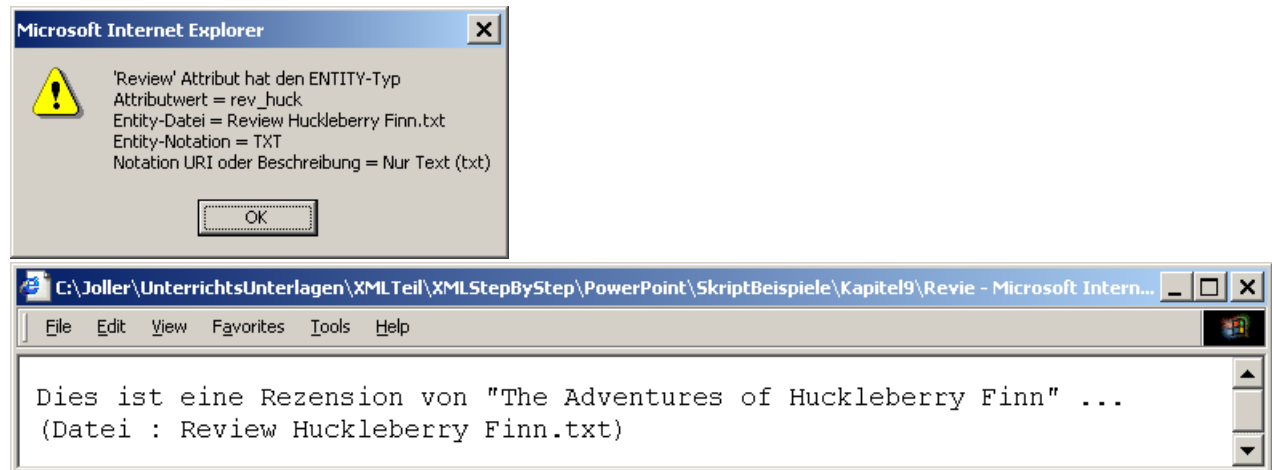
- Script
  - Document = domInventory.XMLDocument;  
Attribute = Document.documentElement.childNodes(0).attributes(0);  
if (Attribute.dataType == "entity") {  
    DisplayText = "" + Attribute.nodeName  
                    + " Attribut hat den ENTITY-Typ" + "\n";  
    DisplayText += "Attributwert = " + Attribute.nodeValue + "\n";  
    Entity =  
    Document.doctype.entities.getNamedItem(Attribute.nodeValue);  
    DisplayText += "Entity-Datei = „  
    + Entity.attributes.getNamedItem("SYSTEM").nodeValue+"\n";  
    NotationName=  
    Entity.attributes.getNamedItem("NDATA").nodeValue;  
    DisplayText += "Entity-Notation = " + NotationName + "\n";  
    Notation =  
    Document.doctype.notations.getNamedItem(NotationName);  
    DisplayText += "Notation URI oder Beschreibung = " +  
    Notation.attributes.getNamedItem("SYSTEM").nodeValue  
    + "\n";  
    alert (DisplayText);  
    location.href = Entity.attributes.getNamedItem("SYSTEM").nodeValue; }  
}

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Anwendungsbeispiel: Kombination Von *Attribute* Und *attributes* Und DTD

- Ausgabe
  - Dateien : *DOM Inventory Entity.htm* und *Inventory Entity.xml*



## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Anwendungsbeispiel: Analyse Eines Dokuments

- Ziel

- Durchsuchen des gesamten DOM Baumes

- Vorgehen

- Mit einem JavaScript wird rekursiv der gesamte DOM Baum analysiert
- Mit Hilfe des DIV-Tags wird das Ergebnis als HTML-Seite angezeigt

```
» <SCRIPT LANGUAGE="JavaScript" FOR="window" EVENT="ONLOAD">
 /* Document-Knoten ermitteln: */
 Document = domXML.XMLDocument;
 /* Mit der Übergabe des Document-Knotens an DisplayNodes beginnen: */
 DisplayDIV.innerHTML = DisplayNodes(Document, 0);
```

```
function DisplayNodes (Node, IndentLevel)
{
 /* Lokale Variablen für Rekursion deklarieren: */
 var i;
 var DisplayString = "";
 /* ... Eigentliches Skript */
```



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Anwendungsbeispiel: Analyse Eines Dokuments

### – Skript Erläuterungen

- die Funktion führt folgende Schritte aus
  - Sie speichert Leerzeilen, deren Anzahl durch den Parameter IndentLevel bestimmt wird

```
» /* Einrückung für diese Ebene erstellen: */
 Indent = "";
 IndentDelta = " ";
 for (i=0; i < IndentLevel; ++i)
 Indent += IndentDelta;
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Anwendungsbeispiel: Analyse Eines Dokuments

### – Skript Erläuterungen

- Sie speichert Anzeigedaten des aktuellen Knotens (Node-Parameter der Funktion DisplayNode)

```
» /* Eigenschaften des aktuellen Knotens anzeigen: */
 DisplayString += Indent + "nodeName: ,,
 + Node.nodeName + "\n,,
 + Indent + "nodeTypeType: ,,
 + Node.nodeType + "\n,,
 + Indent + "nodeTypeString: ,,
 + Node.nodeTypeString + "\n,,
 + Indent + "nodeValue: ,,
 + Node.nodeValue + "\n\n";
```

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Anwendungsbeispiel: Analyse Eines Dokuments

#### – Skript Erläuterungen

- Sie zeigt Informationen zu den untergeordneten Attributes-Knoten des aktuellen Knotens an (eine Ebene weiter eingerückt)

```
» /* Untergeordnete Attribute-Knoten jedes Knotens anzeigen: */
 Indent += IndentDelta;
 for (i=0;
 Node.attributes != null
 && i < Node.attributes.length;
 ++i)
 DisplayString += Indent + "nodeName: „
 + Node.attributes(i).nodeName + "\n„
 + Indent + "nodeTypeType: „
 + Node.attributes(i).nodeType + "\n„
 + Indent + "nodeTypeString: „
 + Node.attributes(i).nodeTypeString
 + "\n„
 + Indent + "nodeValue: „
 + Node.attributes(i).nodeValue
 + "\n\n";
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## **Anwendungsbeispiel: Analyse Eines Dokuments**

### – Skript Erläuterungen

- Sie speichert Informationen zu jedem anderen untergeordneten Knoten, der nicht vom Typ Attribute ist, indem sie sich selbst für jeden dieser Knoten aufruft (rekursiver Funktionsaufruf)

```
» /* Andere untergeordnete Knoten jedes Knotens anzeigen: */
 for (i=0; i < Node.childNodes.length; ++i)
 DisplayString +=
 DisplayNodes (Node.childNodes(i), IndentLevel + 1);
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Anwendungsbeispiel: Analyse Eines Dokuments**

– Skript Erläuterungen

– Sie endet mit der Rückgabe der Zeichenkette, welche sämtliche Knoteninformationen enthält

» `/* Zeichenkette mit den Ergebnissen zurückgeben: */  
return DisplayString;`

## XML-Dokumente Im Internet Darstellen

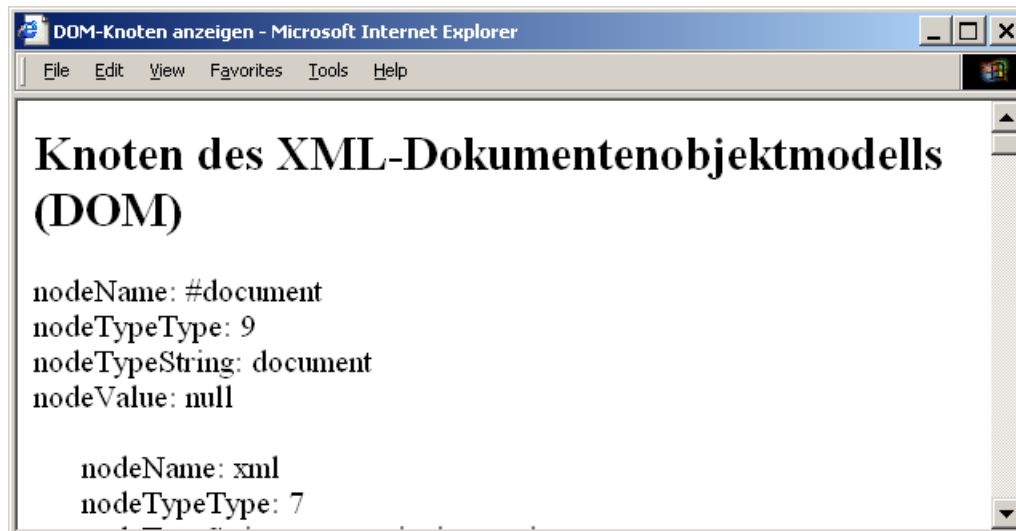
## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Anwendungsbeispiel: Analyse Eines Dokuments

### – Anzeige

- Dateien : *Inventory DOM.xml* und *DOM Show Nodes.htm*

```
» <?xml version="1.0"?>
 <!-- Dateiname: Inventory DOM.xml -->
 <INVENTORY>
 <BOOK Binding="Taschenbuch">
 <TITLE>The Adventures of Huckleberry Finn</TITLE>
 <AUTHOR Born="1835">Mark Twain</AUTHOR>
```

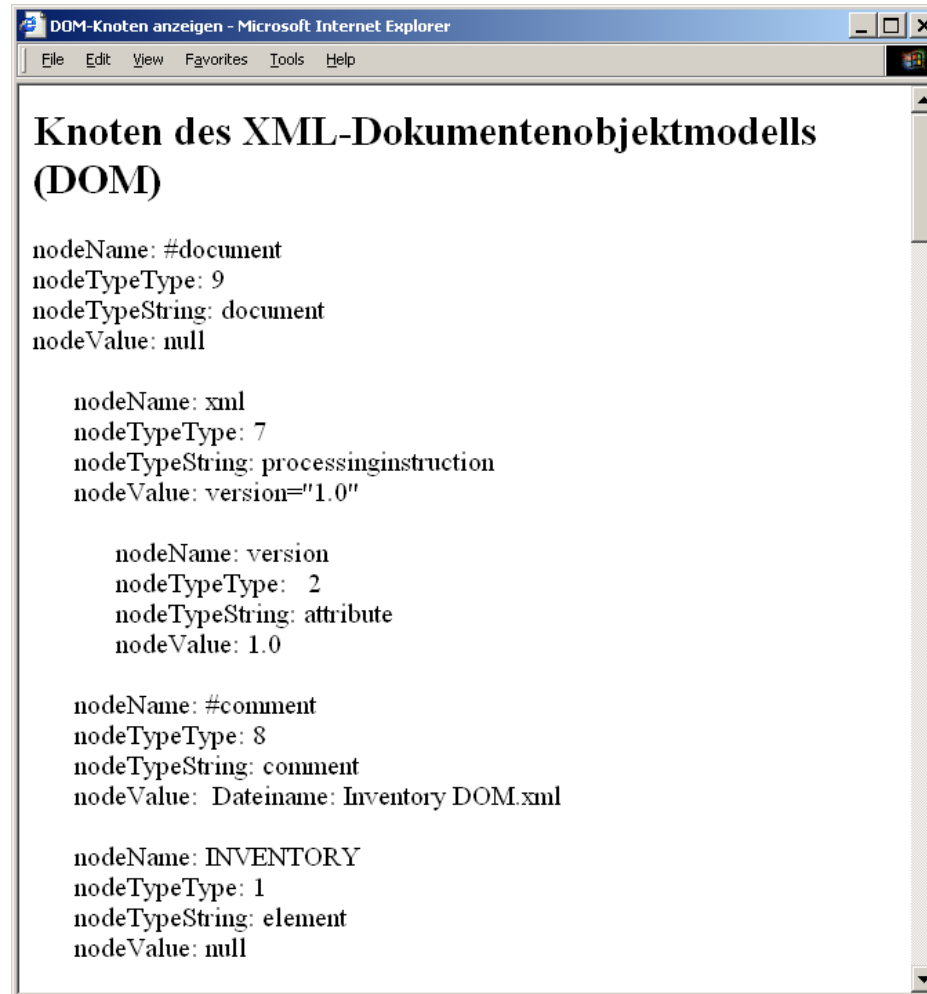


# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Anwendungsbeispiel: Analyse Eines Dokuments

#### – Anzeige



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Anwendungsbeispiel: Gültigkeit Eines Dokuments Prüfen**

- Ziel

- Wir wollen mithilfe vom DOM die Gültigkeit eines XML-Dokuments überprüfen

- Idee

- Falls DOM einen Fehler meldet, kann das Dokument nicht wohlgeformt oder gültig sein

- Falls das XML-Dokument eine DTD (oder XSL) besitzt, also eine fest vorgeschriebene Syntax, dann kann geprüft werden, ob das Dokument gültig (gegenüber dieser Syntax) ist

- Sonst kann nur die Wohlgeformtheit überprüft werden

- Falls das XML-Dokument nicht fehlerfrei ist, werden in der HTML-Seite keine Daten angezeigt

- Ob alle Daten geladen werden, können wir über den *readyState* abfragen



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

**Anwendungsbeispiel: Gültigkeit Eines Dokuments Prüfen**

– Die Lösung

- Skript und Erläuterung dazu

- **<SCRIPT LANGUAGE="JavaScript" FOR="window" EVENT="ONLOAD">**

- » Das Skript wird beim Öffnen des Dokuments ausgeführt

- **Document = domTest.XMLDocument;**

- » Die Wurzel des Dokuments, der *Document*-Knoten wird ermittelt

- **if (Document.readyState == 4)**

- DisplayError ();**

- else**

- Document.onreadystatechange = DisplayError;**

- » Falls *readyState*=4 ist, wurden alle Daten geladen und verarbeitet  
Sonst wird durch die Zuweisung der Funktion *DisplayError* an *onreadystatechange* bewirkt, das immer dann, wenn sich *readyState* ändert, die Funktion *DisplayError* aufgerufen wird.

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Anwendungsbeispiel: Gültigkeit Eines Dokuments Prüfen

### – Die Lösung

- Skript und Erläuterung dazu

```
– function DisplayError () {
 if (Document.readyState != 4)
 return;
 message = "parseError.errorCode: „
 + Document.parseError.errorCode + "\n„
 + "parseError.filepos: „
 + Document.parseError.filepos + "\n„
 + "parseError.line: " + Document.parseError.line
 + "\n„
 + "parseError.linepos: „
 + Document.parseError.linepos + "\n„
 + "parseError.reason: „
 + Document.parseError.reason + "\n„
 + "parseError.srcText: „
 + Document.parseError.srcText + "\n„
 + "parseError.url: " + Document.parseError.url;
 alert (message);
}
```

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

## Anwendungsbeispiel: Gültigkeit Eines Dokuments Prüfen

### – Die Lösung

- Skript und Erläuterung dazu

```
– function DisplayError () {
 if (Document.readyState != 4)
 return;
 ...
}
```

» Falls readyState !=4 ist, sind nicht alle Daten geladen und verarbeitet

```
– function DisplayError () {
 ...
 message = "parseError.errorCode: „
 + Document.parseError.errorCode + "\n„
 + "parseError.filepos: „
 + Document.parseError.filepos + "\n„
 ...
 + "parseError.url: " + Document.parseError.url;
 alert (message);
}
```

» Sonst werden die Fehlercodes abgefragt und anschliessend angezeigt

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von Dom-Skripts Anzeigen

### Anwendungsbeispiel: Gültigkeit Eines Dokuments Prüfen

#### – Anzeige

- Dateien : DOM XML Validity.htm, Inventory.xml



- Dateien DOM XML Validity Fehler.html, Inventory Fehler.xml

