

Tritt nun ein Fehler auf, wird dieser gezielt ausgegeben, wofür PHP die Variable `$php_errormsg` bereitstellt:

```
echo "Fehler: " . $php_errormsg;
```

Zum Umgang mit `$php_errormsg` finden Sie Hinweise in [➔](#) Abschnitt *Die Fehlervariable `$php_errormsg`* auf Seite 358.

```
Fehler: Element book content check failure
```

Abbildung 9.13:  
Fehler bei der  
Validierung

Das Schema verlangte bei diesem Test, dass innerhalb eines Elements mit dem Namen `<book>` ein Element mit dem Namen `<isbn>` auftauchen muss. Geschrieben wurde aber `<Isbn>`, um einen Fehler zu provozieren.

Leider wird – auch in Anbetracht der möglichen Anzahl von Fehlern – nur eine recht unbefriedigende Nachricht erzeugt. Die Meldung besagt nichts weiter, als dass der verlangte Inhalt des `<book>`-Elements nicht vorliegt. Was falsch ist, fehlende oder falsch formatierte Elemente, wird nicht weiter untersucht. Auch eine Angabe über die Fehlerquelle, also Zeile und Spalte, fehlt völlig.

Nur wenige  
Informationen

### 9.3.5 SimpleXML

*SimpleXML* ist eine Ergänzung der DOM-Bibliothek und existiert weitgehend unabhängig von dieser. Es ist eine sehr einfache, kompakte Bibliothek für den schnellen Zugriff auf XML. Durch eine Kompatibilitätsschnittstelle zu DOM kann man schnell von dem einen auf das andere System übergehen.

#### Einfacher Zugriff auf XML-Elemente

Das folgende Beispiel zeigt, wie einfach es ist, Daten mit *SimpleXML* auszulesen:

```
<?php
$xml = simplexml_load_file('data\articles.xml');
foreach($xml->item as $item)
{
    echo "{$item->title} <br>";
}
?>
```

Die Funktion `simplexml_load_file` liest die XML-Datei ein. Dann kann mit einfachen Mitteln ein objektorientierter Zugriff auf die Elemente erfolgen:

```
$item->title
```

Listing 9.30:  
*simplexml\_read*:  
Zugriff auf XML in  
der einfachsten Form

Jedes Element wird als Eigenschaft des darüberliegenden abgebildet. Ein Blick in die entsprechende Struktur der hier benutzten XML-Datei zeigt, wie das gemeint ist:

```
<articles>
  <item>
    <title>PHP Weekly: Issue # 172</title>
    <link>http://www.zend.com/zend/week/week172.php</link>
  </item>
  <item>
    <title>Tutorial: Develop rock-solid code in PHP: Part
      three
    </title>
    <link>http://www.zend.com/zend/tut/tut-hatwar3.php</link>
  </item>
```

### Besonderheiten des Zugriffs auf Elemente und Attribute

Falls Attribute verwendet werden, stehen diese über die Array-Syntax zur Verfügung. Angenommen, die XML-Datei enthält folgendes Element:

```
<title id="3543">
```

Dann würde der Zugriff auf das Attribut *id* folgendermaßen gelingen:

```
$item->title['id']
```

Dieselbe Syntax wird auch verwendet, um gezielt auf bestimmte Elemente der Hierarchie zuzugreifen. So erhalten Sie den ersten Titel des zweiten *<item>*-Elements mit folgendem Code:

```
echo $xml->item[1]->title[0];
```

#### Nullbasierte Indizes

Beachten Sie, dass die Indizes nullbasiert sind. Folgendes wäre übrigens falsch:

```
echo $xml->item->title[1];
```

Dies ergibt nicht, wie es bei einer vergleichbaren XPath-Anweisung der Fall wäre, den zweiten Titel im Dokument, weil *\$xml->item* die Auswahl bereits auf das erste *<item>*-Element beschränkt hat. Freilich heißt dies nicht, dass Sie auf XPath verzichten müssen. *SimpleXML* bietet auch dafür eine passende Methode an.

#### XPath

In [➔](#) Abschnitt 9.4 *XML mit XPath abfragen* ab Seite 796 finden Sie detaillierte Informationen zu XPath.

*Listing 9.31:*  
*simplexml\_xpath:*  
*XPath mit*  
*SimpleXML nutzen*

```
<?php
$xml = simplexml_load_file('data\articles.xml');
foreach($xml->xpath('/articles/item/title') as $title)
{
    echo "{$title} <br>";
}
?>
```

`xpath` ist eine Methode, die XPath-Anweisungen analysiert, ausführt und eine Knotenliste zurückgibt, bei der es sich praktischerweise um ein simples Array handelt, dessen Elemente Objekte vom Typ `SimpleXMLElement` sind.

### SimpleXMLElement näher betrachtet

`SimpleXMLElement` bietet einen objektorientierten Zugriff auf Elemente. Außerdem werden folgende Methoden bereitgestellt:

Methoden	Bedeutung
<code>asXML</code>	Gibt den Inhalt als XML-Zeichenkette zurück
<code>xpath</code>	Führt eine XPath-Anweisung aus und gibt ein Array mit <code>SimpleXMLElement</code> -Objekten zurück
<code>attributes</code>	Gibt eine Auflistung der Attribute zurück
<code>children</code>	Auflistung der Kindelemente

Tabelle 9.10:  
Methoden der Klasse  
`SimpleXMLElement`

Die Eigenschaften der Klasse werden durch die Elemente gebildet, hängen also vom XML-Dokument ab. Dabei werden jeweils die Namen der Elemente benutzt.

### Daten mit SimpleXML schreiben

Das folgende Beispiel zeigt, dass sich auch Daten in XML-Dokumenten ändern lassen:

```
<?php
$xml = simplexml_load_file('data\articles.xml');
$node = $xml->xpath('/articles/item[position]=1');
$node[0]->title = 'PHP 5 - New Release';
echo htmlspecialchars($xml->asXML());
?>
```

Listing 9.32:  
`simplexml_write`:  
Daten in einem  
XML-Dokument  
verändern

Dieses Skript wählt den ersten `<item>`-Knoten über eine XPath-Anweisung aus. Dann wird dessen Titel verändert:

```
$node[0]->title = 'PHP 5 - New Release';
```

Anschließend wird der Inhalt des so modifizierten Dokuments ausgegeben. Für das Speichern auf dem Datenträger benutzen Sie die bekannten Datei-Funktionen. Die Daten erhalten Sie mit `asXML`.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<articles>
  <item>
    <title id="23">PHP 5 - New Release</title>
    <link>http://www.zend.com/zend/week/week172.php</link>
  </item>
  <item>
    <title id="456">Tutorial: Develop rock-solid code in PHP: Part three</title>
    <link>http://www.zend.com/zend/tut/tut-hatwar3.php</link>
  </item>
</articles>
```

Abbildung 9.14:  
Mit `SimpleXML`  
modifiziertes  
Dokument

### Interoperabilität mit DomXML

SimpleXML reicht oft aus, aber manchmal braucht man die gesamte Leistungsfähigkeit der DOM-Bibliothek. Dank zweier einfacher Funktionen kann man aus der einen in die andere Welt gelangen und so parallel mit beiden Bibliotheken arbeiten:

- `simplexml_import_dom($dom)`  
Diese Funktion importiert ein Dokument *\$dom* vom Typ `DomDocument` in die SimpleXML-Welt.
- `dom_import_simplexml($xml)`  
Diese Funktion übergibt ein SimpleXML-Objekt nach DOM. Der Rückgabewert der Funktion hat den Typ `DomDocument`, *\$xml* ist vom Typ `SimpleXMLElement`.

### 9.3.6 XML, HTML und XHTML

Die Welt von XML ist klar, schön und rein. Die Realität leider nicht. Wohlgeformte Dokumente sind im Web eher selten anzutreffen. Die XML-Bibliothek *libxml2* verfügt deshalb über einige sehr interessante Funktionen, die die Brücke zwischen XML und HTML schließen. Dafür stellt die Klasse `DomDocument` folgende Methoden zur Verfügung:

- `LoadHTML`  
Lädt ein HTML-Dokument aus einer Zeichenkette.
- `LoadHTMLFile`  
Lädt ein HTML-Dokument aus einer Datei. Hier sind alle üblichen Wrapper erlaubt, sodass auch Dokumente von anderen Servern per HTTP oder FTP geladen werden können.
- `SaveHTML`  
Speichert das XML-Dokument im nicht wohlgeformten aber HTML 4-Standard kompatiblen Format als Zeichenkette.
- `SaveHTMLFile`  
Speichert das XML-Dokument im nicht wohlgeformten aber HTML 4-Standard kompatiblen Format als Datei. Hier sind mit Einschränkungen einige Wrapper erlaubt, sodass auch Dokumente auf andere Server beispielsweise mit FTP gespeichert werden können.

Das folgende Beispiel lädt eine nicht wohlgeformte (und damit typische) HTML-Datei und gibt sie in drei Versionen aus:

1. Unverändert
2. Als standardkonforme HTML-Datei
3. Als wohlgeformtes XML-Dokument, mithin also als XHTML