

Web-Informationssysteme

XPath & XQuery

5. Februar 2010

Quelle: <http://www.w3schools.com/XPath/>

1 XPath Knoten

Beispiel XML-Dokument:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book>
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

Die Repräsentation eines XML Dokuments für XPath ist ein Baum, dessen oberster Knoten, der "virtuelle Root", das gesamte Dokument repräsentiert.

Beispiele für Knoten sind:

```
<bookstore> (document element node)
<author>J K. Rowling</author> (element node)
lang="en" (attribute node)
```

Beispiel für atomare Werte (Attribute):

```
J K. Rowling
"en"
```

Elemente sind atomare Werte oder Knoten.

Beziehungen von Knoten

Parent:

Jedes Element hat einen Vater.

Das Buch-Element ist das Vater-Element (parent) von title, author, year und price.

```
<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

Children:

Elementknoten können keinen, einen oder mehrere Kinder haben.

title, author, year und price sind Kinder (children) von dem Buch-Element.

```
<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

Sibling:

Knoten die den selben Vater haben.
title, author, year und price sind Geschwister (siblings).

```
<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

Ancestors:

Knoten die einen Vater, oder den Vater vom Vater, etc. haben (i.A. Vorfahren).
Das Buch-Elemente, das Buchladen-Element und der virtuelle Root sind die Vorfahren (ancestors) von title.

```
<bookstore>

<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

</bookstore>
```

Descendants:

Knoten die einen Sohn, oder den Sohn vom Sohn, etc. haben (i.A. Nachfahren).
Die Nachfahren (descendants) vom Buchladen-Element sind book, title, author, year und price .

```
<bookstore>

<book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

</bookstore>
```

2 XPath Syntax

Beispiel:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

<book>
  <title lang="eng">Harry Potter</title>
  <price>29.99</price>
</book>

<book>
  <title lang="eng">Learning XML</title>
  <price>39.95</price>
</book>

</bookstore>
```

Knoten auswählen

Ausdruck	Beschreibung
Knotennamen	Selektiert alle Kinderknoten vom ausgewählten Knoten
/	Selektiert von der Wurzel aus
//	Selektiert Knoten im Dokument ausgehend vom aktuellen Knoten die gematched werden, egal wo die Knoten sind
.	Selektiert den aktuellen Knoten
..	Selektiert den Vaterknoten vom aktuellen Knoten
@	Selektiert Attribute

Beispiel:

Ausdruck	Beschreibung
bookstore	Selektiert alle Buchladen-Kinder des Kontext-Knotens
/bookstore	Selektiert alle Buchladen-Kinder des Wurzel-Elements
bookstore/book	Selektiert alle Buch-Elemente die Kinder (child) vom Buchladen-Element sind
//book	Selektiert alle Buch-Elemente, egal wo sie sind
bookstore//book	Selektiert alle Buch-Elemente die Nachfahren vom Buchladen-Element sind
//@lang	Selektiert alle Attribute, deren label/name 'lang' ist

Prädikate

Ausdruck	Beschreibung
/bookstore/book[1]	Selektiert das erste Buch-Element das Kind vom Buchladen-Element ist
/bookstore/book[last()]	Selektiert das letzte Buch-Element das Kind vom Buchladen-Element ist
/bookstore/book[last()-1]	Selektiert das vorletzte Buch-Element, das Kind eines Buchladen-Elements ist, das selber Kind des virtuellen Roots ist
/bookstore/book[position()<3]	Selektiert die ersten 2 Buch-Elemente, die Kinder vom Buchladen-Element sind
//title[@lang]	Selektiert alle Titel-Elemente, die ein Attribut mit label/name 'lang' haben
//title[@lang='eng']	Selektiert alle Titel-Elemente, die ein Attribut mit label/name 'lang' und wert 'eng' haben
/bookstore/book[price>35.00]	Selektiert alle Buch-Elemente unterhalb des Buchladen-Element die teurer als 35 sind
//title[@lang='eng']	Selektiert alle Titel-Elemente der Buch-Elemente unterhalb des Buchladen-Elements die teurer als 35 sind

Weiter Prädikate:

- Relationszeichen: = != and or < > <= >=
- Zeichenkettenfunktionen:

- `normalize-space()` – Entfernen von Leerzeichen am Anfang und Ende des Strings und Reduktion aufeinanderfolgender Leerzeichen auf eines
 - `substring()` – Einen Teilstring selektieren
 - `substring-before(source, splitter)` – Einen Teilstring vor dem ersten Vorkommen des Trennzeichens selektieren
 - `substring-after(source, splitter)` – Einen Teilstring nach dem ersten Vorkommen des Trennzeichens selektieren
 - `string-length()` – Länge des Strings
- Numerische Funktionen: `+` `-` `*` `div` `[[Modulo|mod]]`
 - Knotenmengen-Funktionen:
 - `count()` – Anzahl der Knoten in einer Knotenmenge
 - `id()` – Selektiert Elemente über die `[[Dokumenttypdefinition|DTD]]`-ID
 - `name()` – Name des Knotens

Unbekannte Knoten auswählen

XPath Wildcards:

Ausdruck	Beschreibung
<code>*</code>	Selektiert alle Elementknoten
<code>/@</code>	Selektiert alle Attributknoten
<code>node()</code>	Selektiert alle Knoten von jedem Typ

Ausdruck	Beschreibung
<code>/bookstore/*</code>	Selektiert alle Kinderknoten vom Buchladen-Element
<code>//*</code>	Selektiert alle Elemente im Dokument
<code>//title[@*]</code>	Selektiert alle title Elemente die ein beliebiges Attribut haben

Mit dem `|` Operator kann man mehrere Pfade auswählen.

3 XPath-Axen

Ausdruck	Beschreibung
ancestor	Selektiert alle Vorfahren (Vater, Grossvater, etc.) vom aktuellen Knoten
ancestor-or-self	Selektiert alle Vorfahren (Vater, Grossvater, etc.) vom aktuellen Knoten und sich selbst
attribute	Selektiert alle Attribute vom aktuellen Knoten
child	Selektiert alle Kinder vom aktuellen Knoten
descendant	Selektiert alle Nachfahren (Kinder, Kinder der Kinder) vom aktuellen Knoten
descendant-or-self	Selektiert alle Nachfahren (Kinder, Kinder der Kinder) vom aktuellen Knoten und sich selbst
following	Selektiert alles in dem Dokument nach dem schliessen-Tag vom aktuellen Knoten
following-sibling	Selektiert alle Geschwister nach dem aktuellen Knoten
namespace	Selektiert alle namespace-Knoten vom aktuellen Knoten
parent	Selektiert den Vater vom aktuellen Knoten
preceding	Selektiert alles in dem Dokument vor dem öffnen-Tag vom aktuellen Knoten
preceding-sibling	Selektiert alle Geschwister vor dem aktuellen Knoten
self	Selektiert den aktuellen Knoten (sich selbst)

4 Location Path Expression

Ein Location Path kann relativ oder absolut sein. Ein absoluter Pfad beginnt mit (/), ein relativer nicht. In beiden Fällen besteht der Location Path aus mehreren Schritten, die mit (/) getrennt werden.

Syntax:

`axisname::nodetest [predicate]`

Ausdruck	Beschreibung
child::book	Selektiert alle Buchknoten, die Kinder vom aktuellen Knoten sind
attribute::lang	Selektiert alle Attribute mit lang vom aktuellen Knoten
child::*	Selektiert alle Kinder vom aktuellen Knoten
attribute::*	Selektiert alle Attribute vom aktuellen Knoten
child::text()	Selektiert alle text-Kinder vom aktuellen Knoten
child::node()	Selektiert alle Kindknoten vom aktuellen Knoten
descendant::book	Selektiert alle Buch-Nachfahren vom aktuellen Knoten
ancestor::book	Selektiert alle Buch-Vorfahren vom aktuellen Knoten
ancestor-or-self::book	Selektiert alle Buch-Vorfahren und sich selber, sofern das ein Buch-Knoten ist
child::price	Selektiert alle Preis Kinder vom aktuellen Knoten