

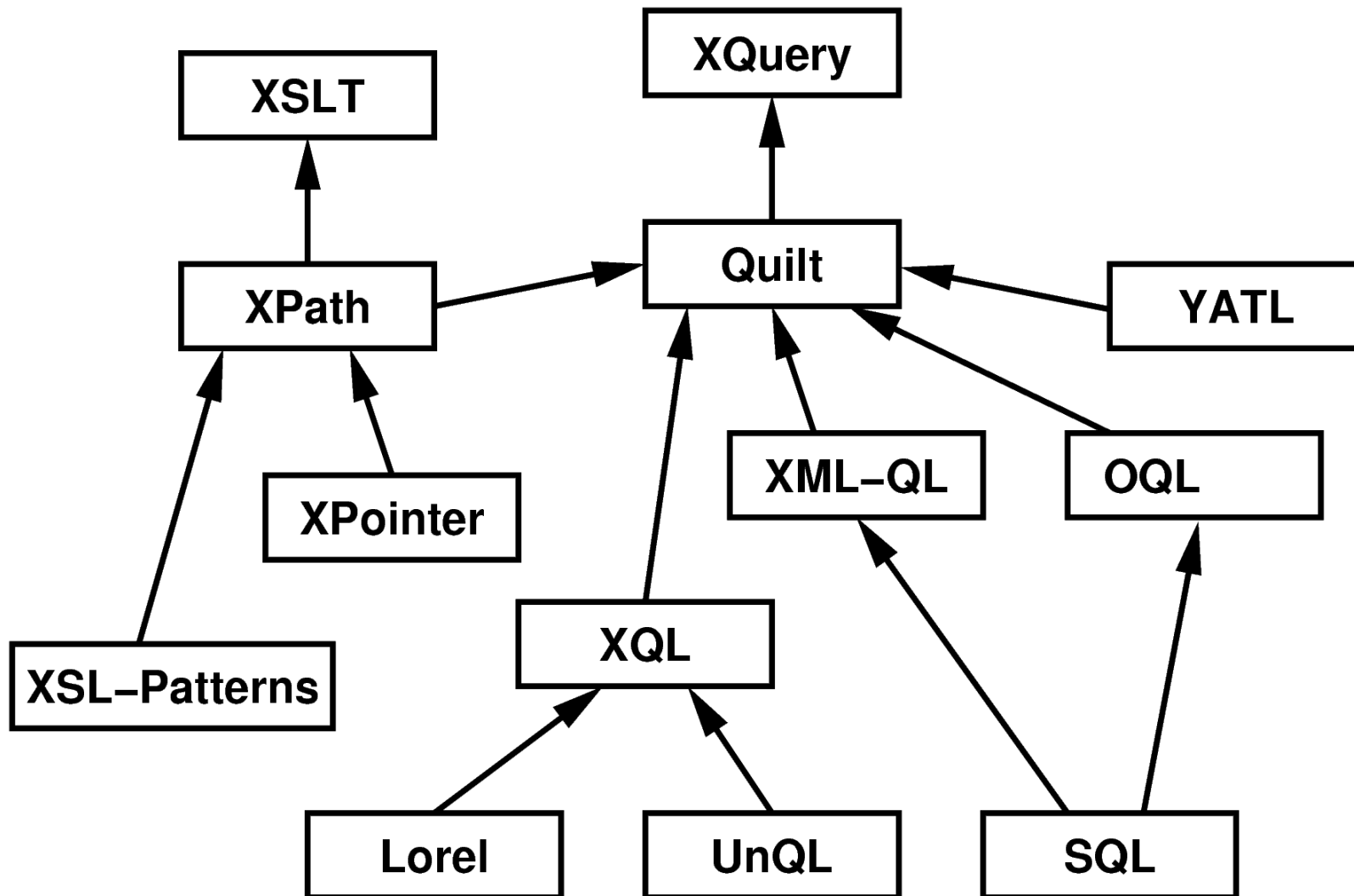
Seminar "*Konzepte von Datenbanken und Informationssystemen*"  
SS 2003

# XPath und XQuery

# Überblick

- **Einführung**  
Vorgeschichte, Entstehung der Sprachen
- **XPath**  
XML Lokalisierungssprache (XPath 1.0)
- **XQuery**  
XML Anfragesprache (XQuery 1.0)
- **Use Cases zu XQuery**
- **Ausblick**  
Zukunft von XQuery, Praxiseinsatz

# XQuery Vorgeschichte



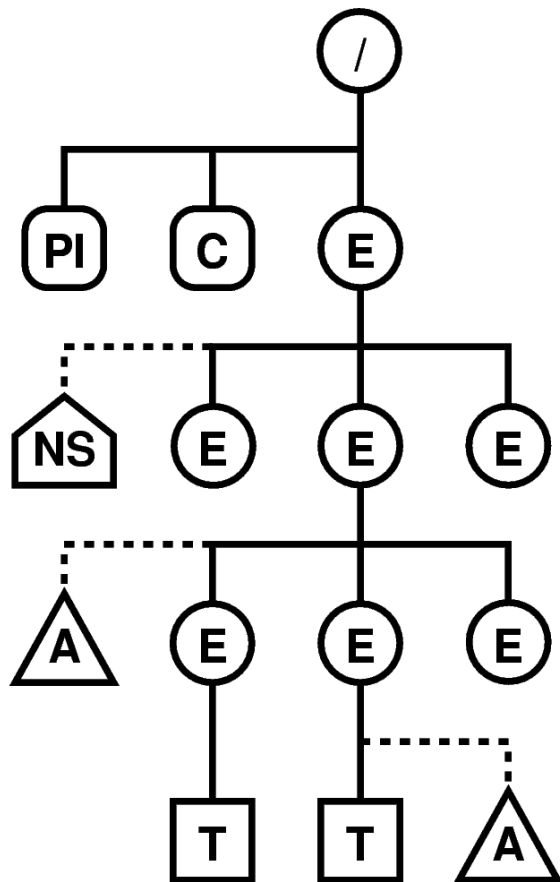
# Überblick

- **Einführung**  
Vorgeschichte, Entstehung der Sprachen
- **XPath**  
XML Lokalisierungssprache (XPath 1.0)
- **XQuery**  
XML Anfragesprache (XQuery 1.0)
- **Use Cases zu XQuery**
- **Ausblick**  
Zukunft von XQuery, Praxiseinsatz

## XPath

- Nicht-XML Syntax
- Addressieren von Teilen eines XML-Dokumentes
- Komponente für anderen Spezifikationen (XPath, XSLT)
- XPath 1.0 seit 1999 Recommendation

## XPath Datenmodell



```

<?xml version="1.0"?>
<?xml-stylesheet href...?>

<!-- Neue Teile 2003 -->
<rs:katalog
  xmlns:rs="http://...">
  <artikel id="#MB-AS23">
    <bezeichnung>
      Kabel f&uuml;r ...
    </bezeichnung>
    <preis waehrung="EUR">
      219.99
    </preis>
  </artikel>
  <artikel id="#LAN-101">
    ...
  </rs:katalog>

```

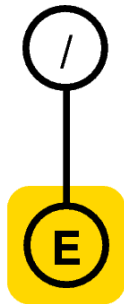
## XPath Datenmodell



```
<?xml version="1.0"?>
<?xml-stylesheet href...?>

<!-- Neue Teile 2003 -->
<rs:katalog
  xmlns:rs="http://...">
  <artikel id="#MB-AS23">
    <bezeichnung>
      Kabel f&uuml;r ...
    </bezeichnung>
    <preis waehrung="EUR">
      219.99
    </preis>
  </artikel>
  <artikel id="#LAN-101">
    ...
  </rs:katalog>
```

## XPath Datenmodell

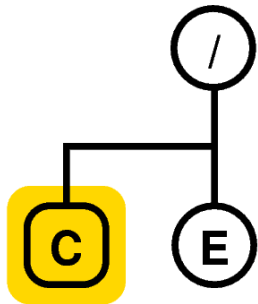


```
<?xml version="1.0"?>  
<?xml-stylesheet href...?>
```

```
<!-- Neue Teile 2003 -->
```

```
<rs:katalog  
  xmlns:rs="http://...">  
  <artikel id="#MB-AS23">  
    <bezeichnung>  
      Kabel f&uuml;r ...  
    </bezeichnung>  
    <preis waehrung="EUR">  
      219.99  
    </preis>  
  </artikel>  
  <artikel id="#LAN-101">  
    ...  
  </rs:katalog>
```

## XPath Datenmodell

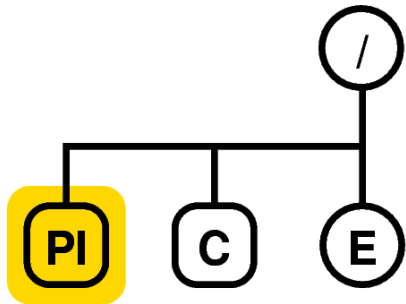


```
<?xml version="1.0"?>  
<?xml-stylesheet href...?>
```

```
<!-- Neue Teile 2003 -->
```

```
<rs:katalog  
  xmlns:rs="http://...">  
  <artikel id="#MB-AS23">  
    <bezeichnung>  
      Kabel f&uuml;r ...  
    </bezeichnung>  
    <preis waehrung="EUR">  
      219.99  
    </preis>  
  </artikel>  
  <artikel id="#LAN-101">  
    ...  
  </rs:katalog>
```

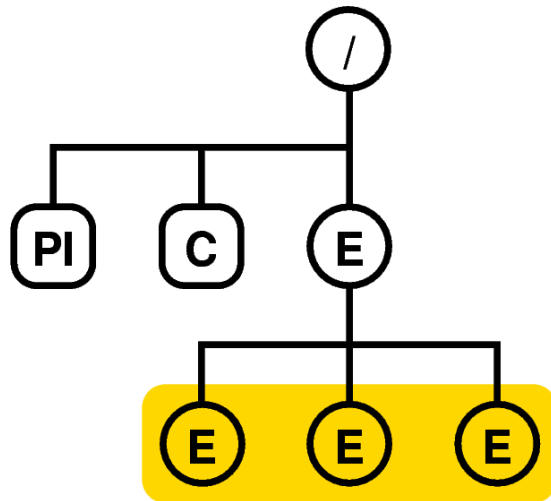
## XPath Datenmodell



```
<?xml version="1.0"?>  
<?xml-stylesheet href...?>
```

```
<!-- Neue Teile 2003 -->  
<rs:katalog  
  xmlns:rs="http://...">  
  <artikel id="#MB-AS23">  
    <bezeichnung>  
      Kabel f&uuml;r ...  
    </bezeichnung>  
    <preis waehrung="EUR">  
      219.99  
    </preis>  
  </artikel>  
  <artikel id="#LAN-101">  
    ...  
  </rs:katalog>
```

## XPath Datenmodell



```
<?xml version="1.0"?>  
<?xml-stylesheet href...?>
```

```
<!-- Neue Teile 2003 -->
```

```
<rs:katalog
```

```
  xmlns:rs="http://...">
```

```
  <artikel id="#MB-AS23">
```

```
    <bezeichnung>
```

```
      Kabel f&uuml;r ...
```

```
    </bezeichnung>
```

```
    <preis waehrung="EUR">
```

```
      219.99
```

```
    </preis>
```

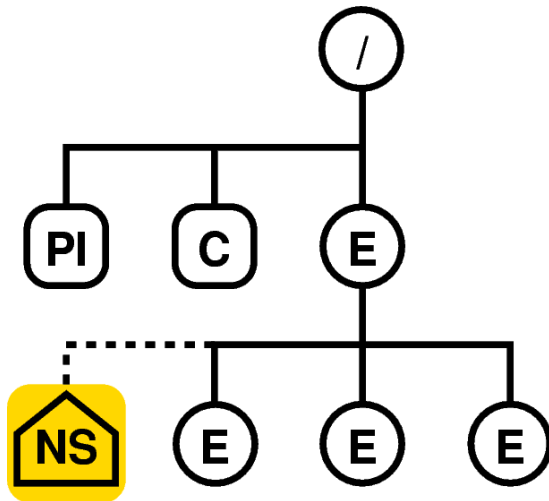
```
  </artikel>
```

```
  <artikel id="#LAN-101">
```

```
    ...
```

```
</rs:katalog>
```

## XPath Datenmodell



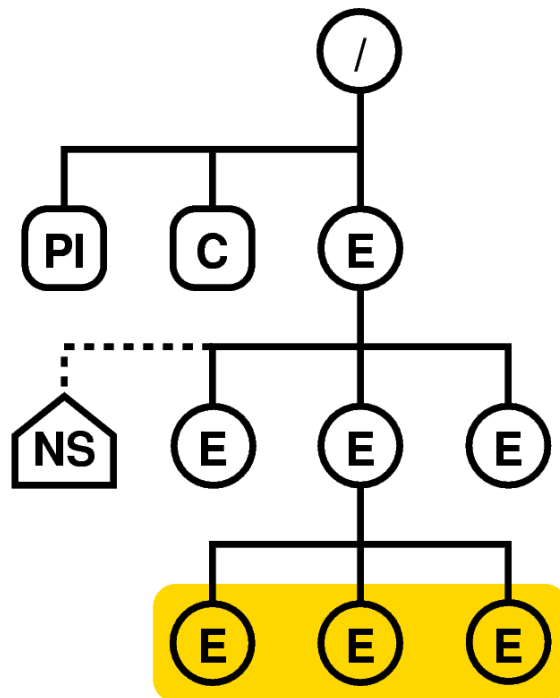
```

<?xml version="1.0"?>
<?xml-stylesheet href...?>

<!-- Neue Teile 2003 -->
<rs:katalog
  xmlns:rs="http://...">
  <artikel id="#MB-AS23">
    <bezeichnung>
      Kabel f&uuml;r ...
    </bezeichnung>
    <preis waehrung="EUR">
      219.99
    </preis>
  </artikel>
  <artikel id="#LAN-101">
    ...
  </rs:katalog>

```

## XPath Datenmodell



```
<?xml version="1.0"?>
<?xml-stylesheet href...?>
```

```
<!-- Neue Teile 2003 -->
```

```
<rs:katalog
```

```
  xmlns:rs="http://...">
```

```
  <artikel id="#MB-AS23">
```

```
    <bezeichnung>
```

```
      Kabel f&uuml;r ...
```

```
    </bezeichnung>
```

```
    <preis waehrung="EUR">
```

```
      219.99
```

```
    </preis>
```

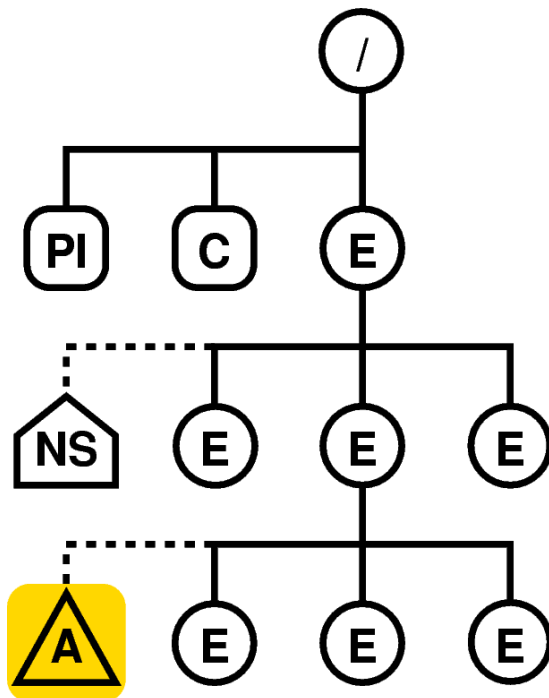
```
  </artikel>
```

```
  <artikel id="#LAN-101">
```

```
    ...
```

```
</rs:katalog>
```

## XPath Datenmodell



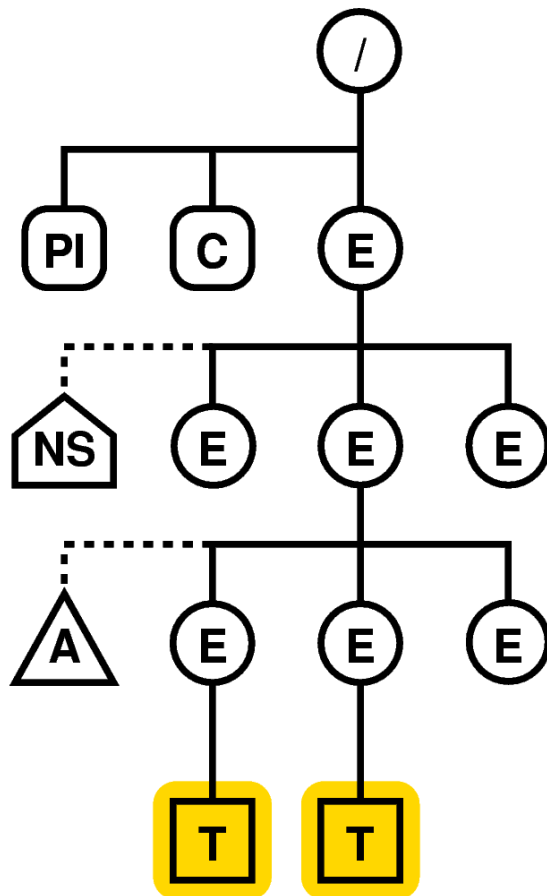
```

<?xml version="1.0"?>
<?xml-stylesheet href...?>

<!-- Neue Teile 2003 -->
<rs:katalog
  xmlns:rs="http://...">
  <artikel id="#MB-AS23">
    <bezeichnung>
      Kabel f&uuml;r ...
    </bezeichnung>
    <preis waehrung="EUR">
      219.99
    </preis>
  </artikel>
  <artikel id="#LAN-101">
    ...
  </rs:katalog>

```

## XPath Datenmodell



```
<?xml version="1.0"?>
<?xml-stylesheet href...?>
```

```
<!-- Neue Teile 2003 -->
```

```
<rs:katalog
```

```
  xmlns:rs="http://...">
```

```
  <artikel id="#MB-AS23">
```

```
    <bezeichnung>
```

```
      Kabel f&uuml;r ...
```

```
    </bezeichnung>
```

```
    <preis waehrung="EUR">
```

```
      219.99
```

```
    </preis>
```

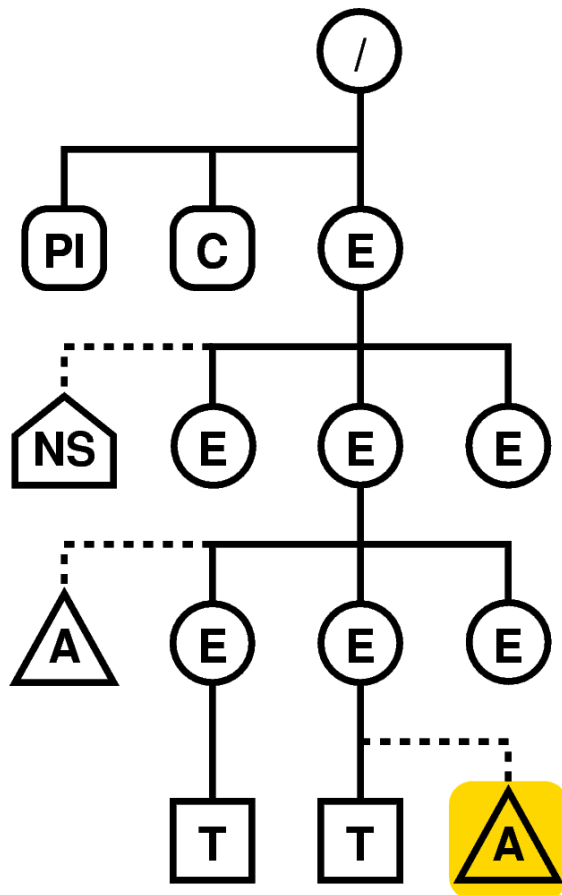
```
  </artikel>
```

```
  <artikel id="#LAN-101">
```

```
    ...
```

```
</rs:katalog>
```

## XPath Datenmodell



```
<?xml version="1.0"?>
<?xml-stylesheet href...?>
```

```
<!-- Neue Teile 2003 -->
```

```
<rs:katalog
```

```
  xmlns:rs="http://...">
```

```
  <artikel id="#MB-AS23">
```

```
    <bezeichnung>
```

```
      Kabel f&uuml;r ...
```

```
    </bezeichnung>
```

```
    <preis waehrung="EUR">
```

```
      219.99
```

```
    </preis>
```

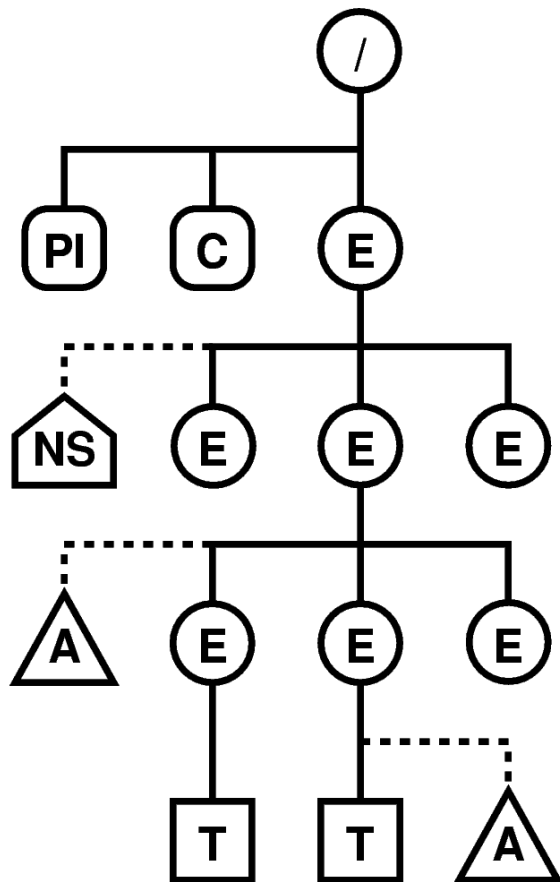
```
  </artikel>
```

```
  <artikel id="#LAN-101">
```

```
    ...
```

```
</rs:katalog>
```

# XPath Datenmodell



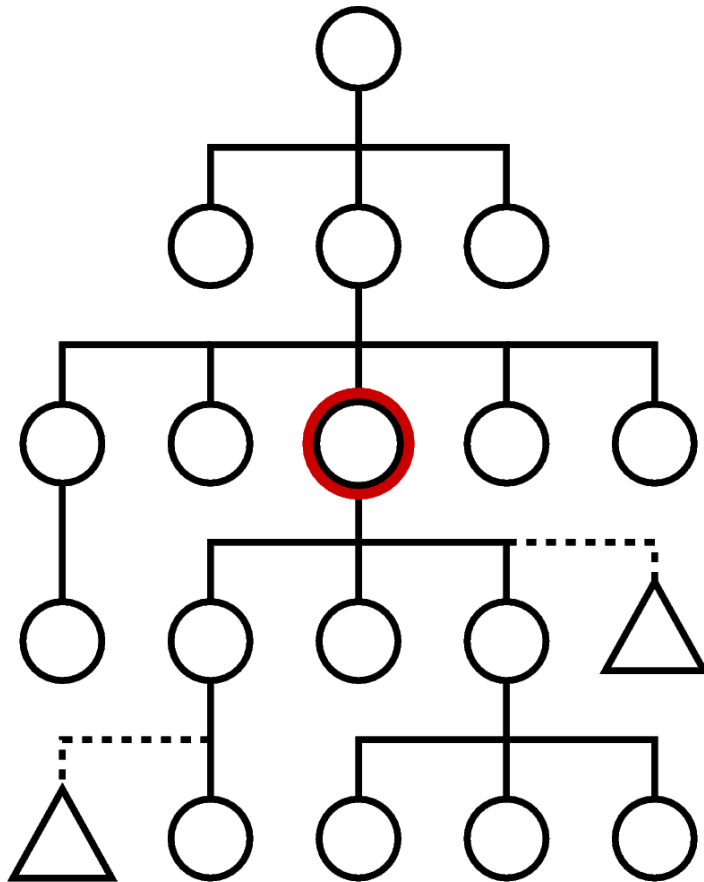
## 7 Knotentypen:

- Wurzelknoten
- Elementknoten
- Attributknoten
- Textknoten
- Kommentarknoten
- Verarbeitungsanweisungsknoten
- Namensraumknoten

## XPath Lokalisierungspfade

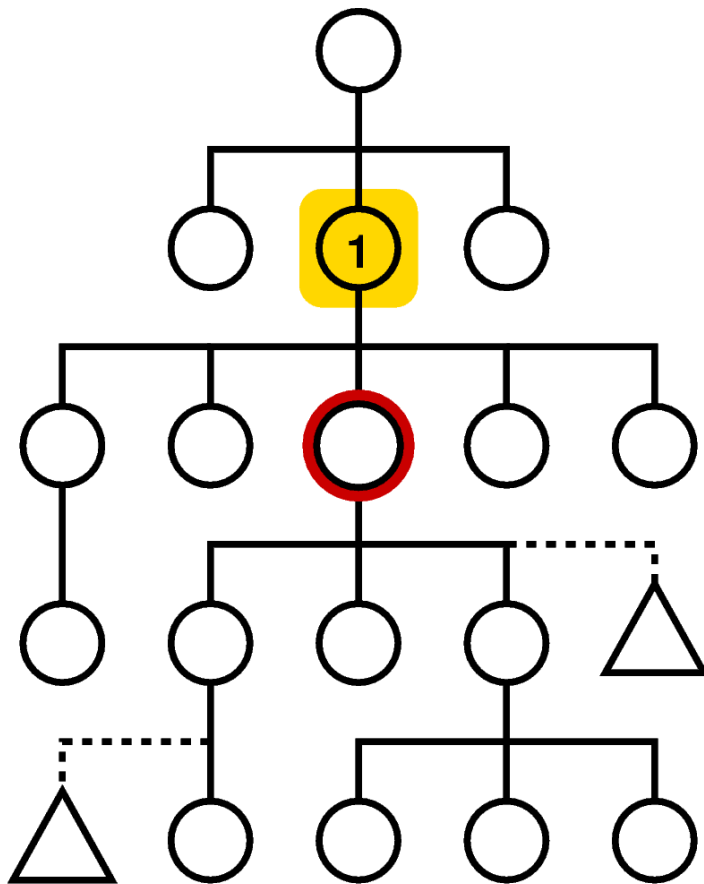
- Pfadausdrücke, um Knotenmengen auszuwählen
- Bezieht sich stets auf einen *Kontextknoten*
- Syntax an UNIX-Shell angelehnt
- Absolute und relative Lokalisierungspfade  
**AbsPfad ::= '/' RelPfad**
- Besteht aus Lokalisierungsschritten  
**RelPfad ::= Schritt ('/' Schritt)\***
- Lokalisierungsschritt besteht aus Achse, Knotentest und Prädikaten  
**Schritt ::= Achse '::' Knotentest Prädikat\***  
| **Kurzschreibweise**

## XPath Achsen



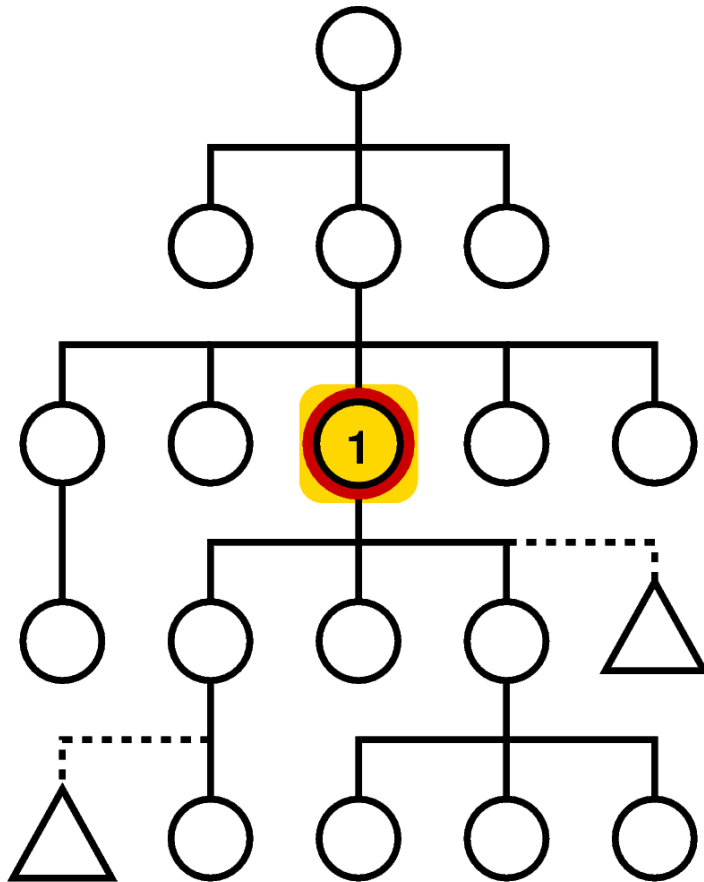
- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- *following-sibling*-Achse
- *preceding*-Achse
- *following*-Achse

## XPath Achsen



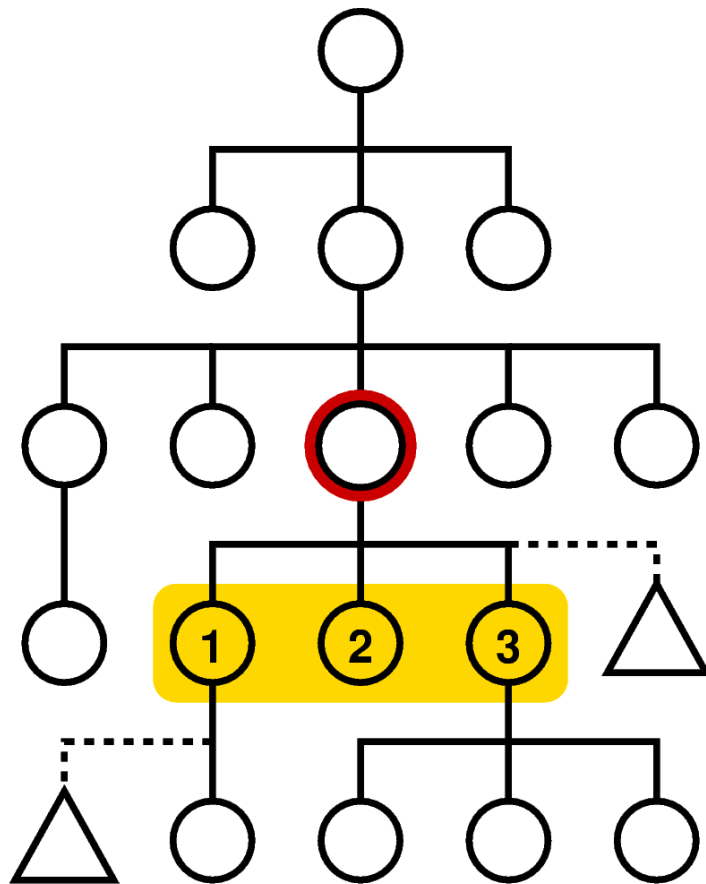
- *parent-Achse*
- *self-Achse*
- *child-Achse*
- *ancestor-Achse*
- *ancestor-or-self-Achse*
- *descendant-Achse*
- *descendant-or-self-Achse*
- *preceding-sibling-Achse*
- *following-sibling-Achse*
- *preceding-Achse*
- *following-Achse*

## XPath Achsen



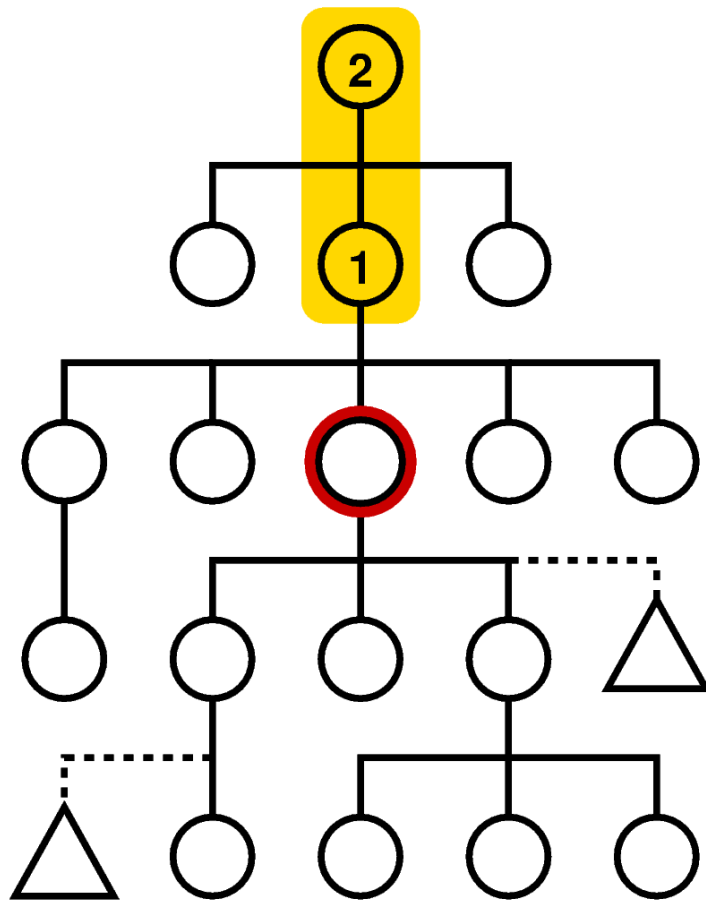
- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- *following-sibling*-Achse
- *preceding*-Achse
- *following*-Achse

## XPath Achsen



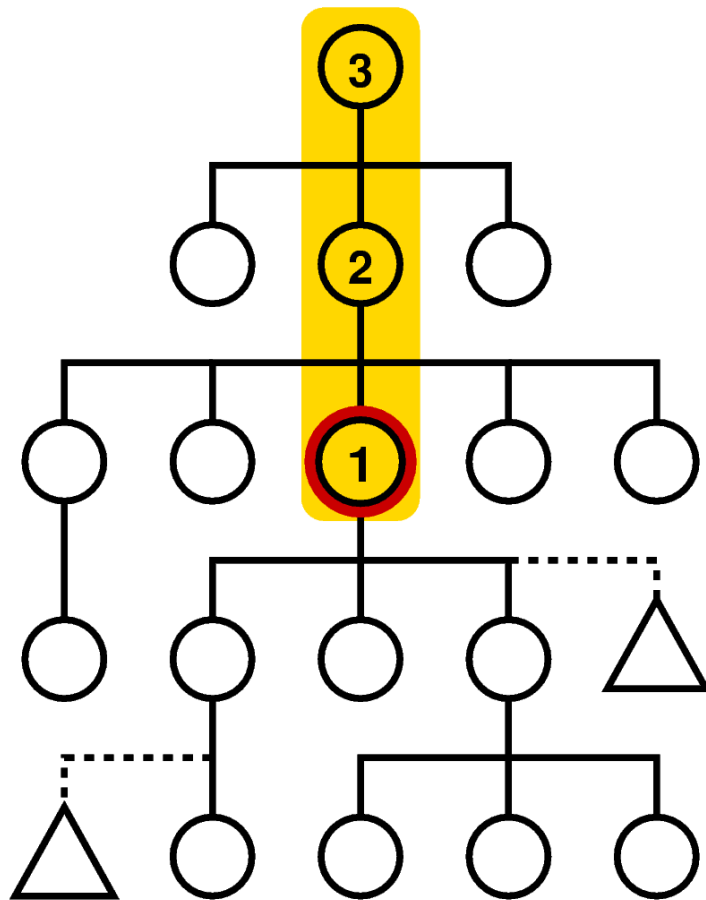
- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- *following-sibling*-Achse
- *preceding*-Achse
- *following*-Achse

## XPath Achsen



- *parent*-Achse
- *self*-Achse
- *child*-Achse
- ***ancestor*-Achse**
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- *following-sibling*-Achse
- *preceding*-Achse
- *following*-Achse

## XPath Achsen

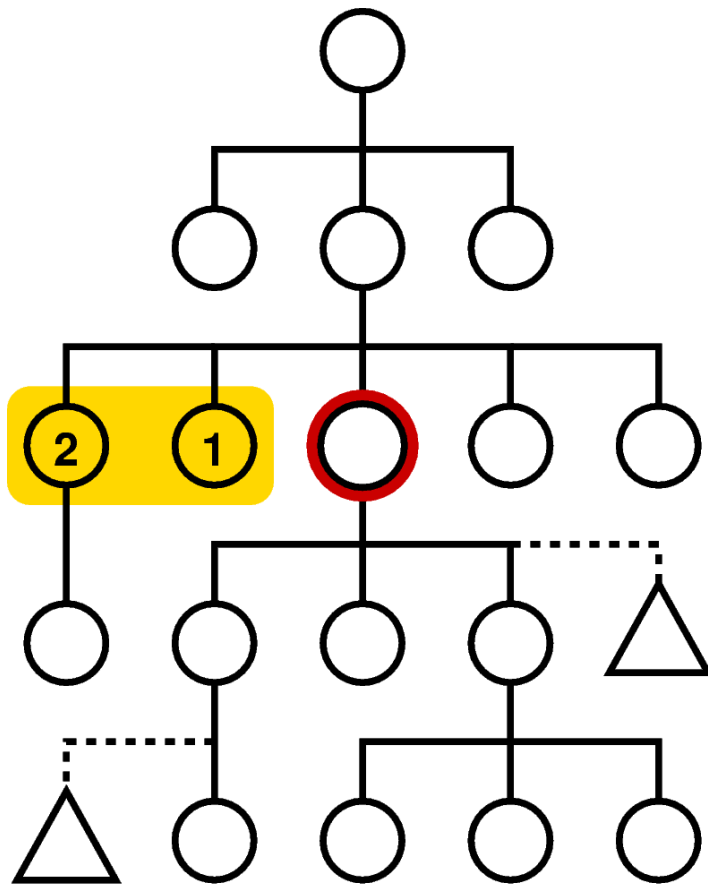


- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- ***ancestor-or-self*-Achse**
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- *following-sibling*-Achse
- *preceding*-Achse
- *following*-Achse



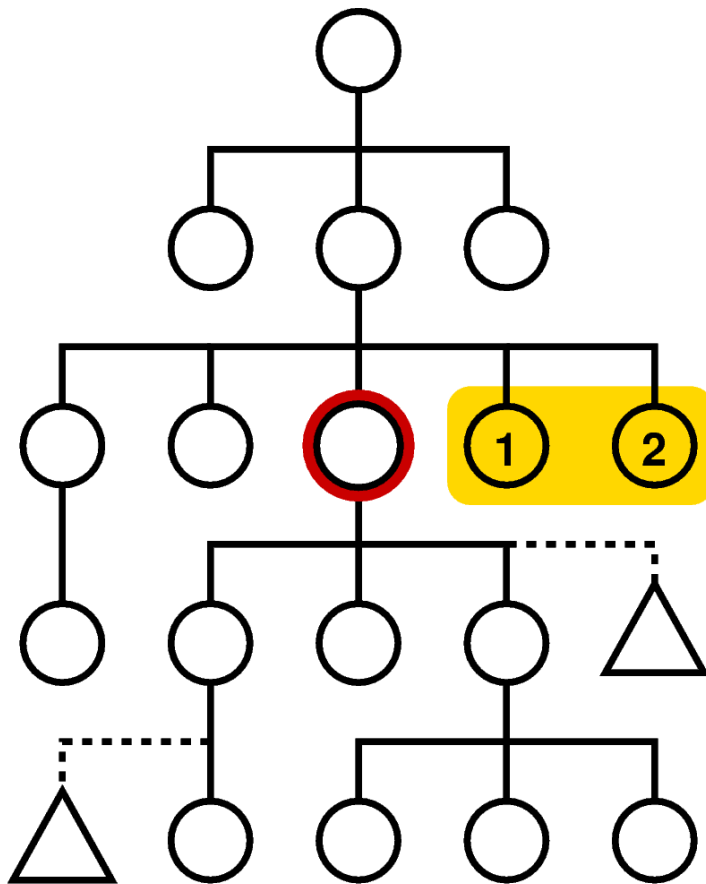


## XPath Achsen



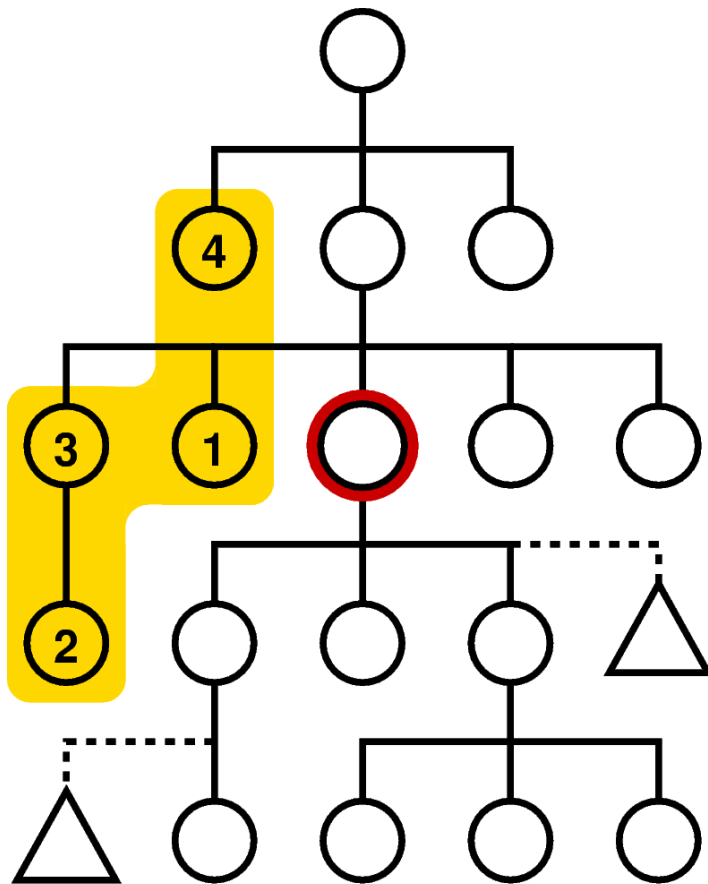
- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- ***preceding-sibling*-Achse**
- *following-sibling*-Achse
- *preceding*-Achse
- *following*-Achse

## XPath Achsen



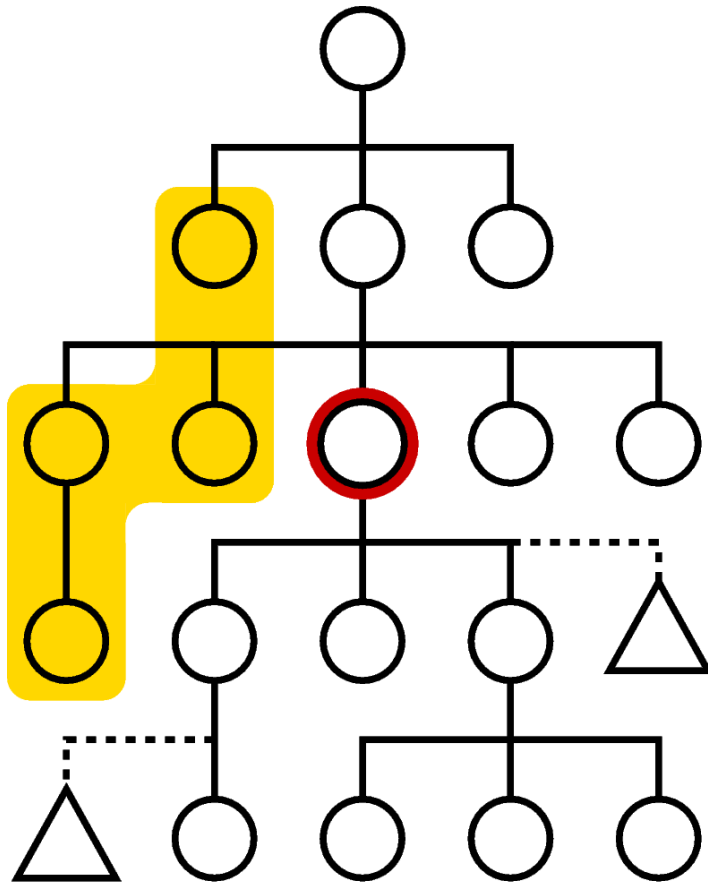
- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- ***following-sibling*-Achse**
- *preceding*-Achse
- *following*-Achse

## XPath Achsen



- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- *following-sibling*-Achse
- ***preceding*-Achse**
- *following*-Achse

## XPath Achsen



```
<!-- Anfang -->
```

```
<katalog >
  <artikel >
    <preis />
  </artikel >
```

```
<artikel />
```

```
<artikel >
  <preis >
```

```
...
```

```
</artikel >
```

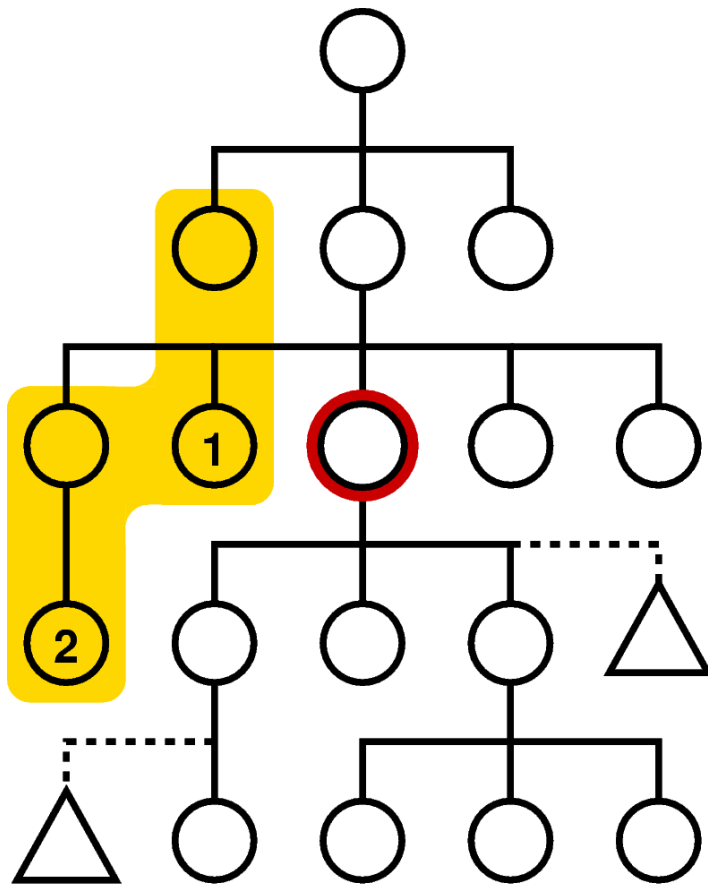
```
...
```

```
</katalog >
```

```
<!-- Ende -->
```



## XPath Achsen



```
<!-- Anfang -->
```

```
<katalog >
  <artikel >
    <preis />
  </artikel >
```

```
<artikel />
```

```
<artikel >
  <preis >
    ...
```

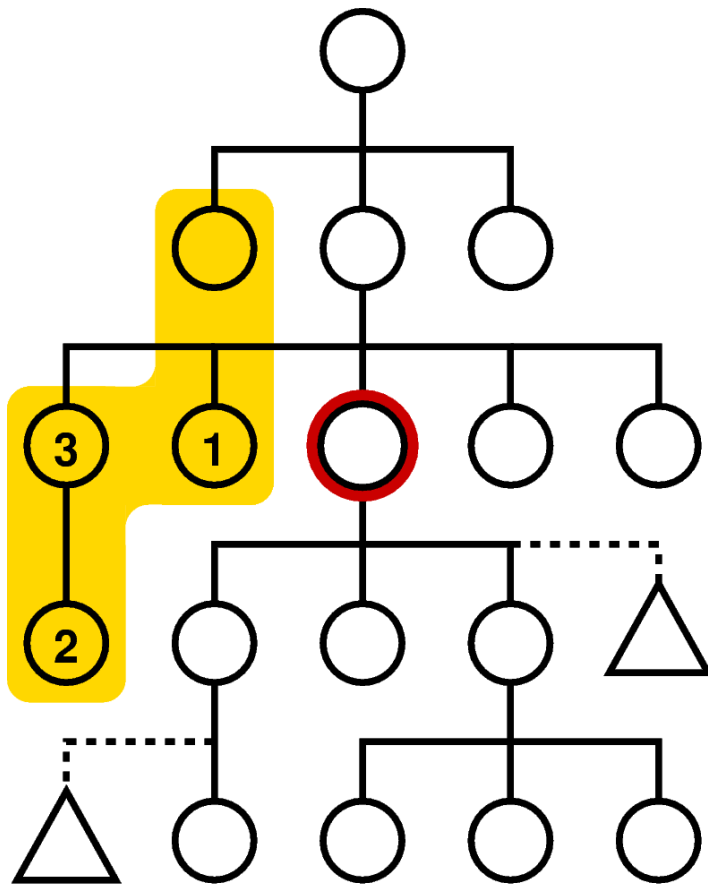
```
</artikel >
```

```
...
```

```
</katalog >
```

```
<!-- Ende -->
```

## XPath Achsen



```
<!-- Anfang -->
```

```
<katalog >
```

```
  <artikel >
```

```
    <preis />
```

```
  </artikel >
```

```
<artikel />
```

```
<artikel >
```

```
  <preis >
```

```
  ...
```

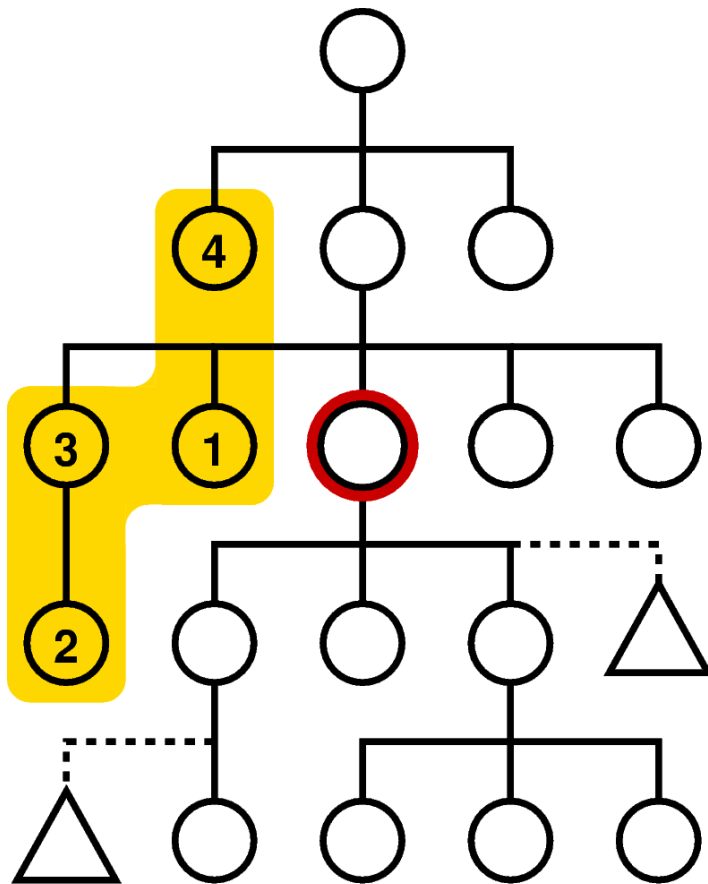
```
</artikel >
```

```
...
```

```
</katalog >
```

```
<!-- Ende -->
```

## XPath Achsen



```
<!-- Anfang -->
```

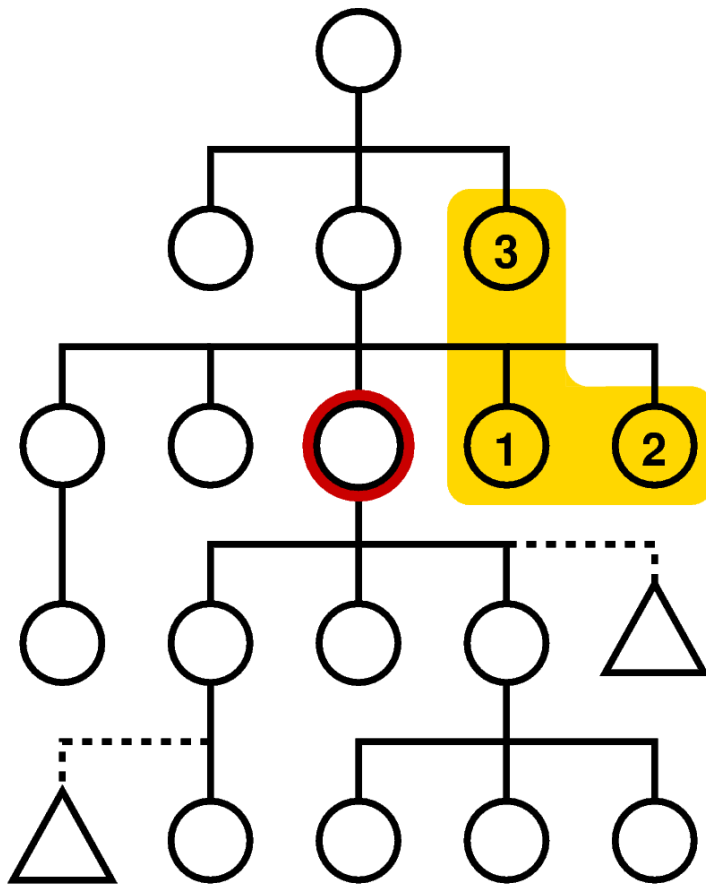
```
<katalog >
  <artikel >
    <preis />
  </artikel >

  <artikel />

  <artikel >
    <preis >
      ...
    </artikel >
    ...
  </katalog >

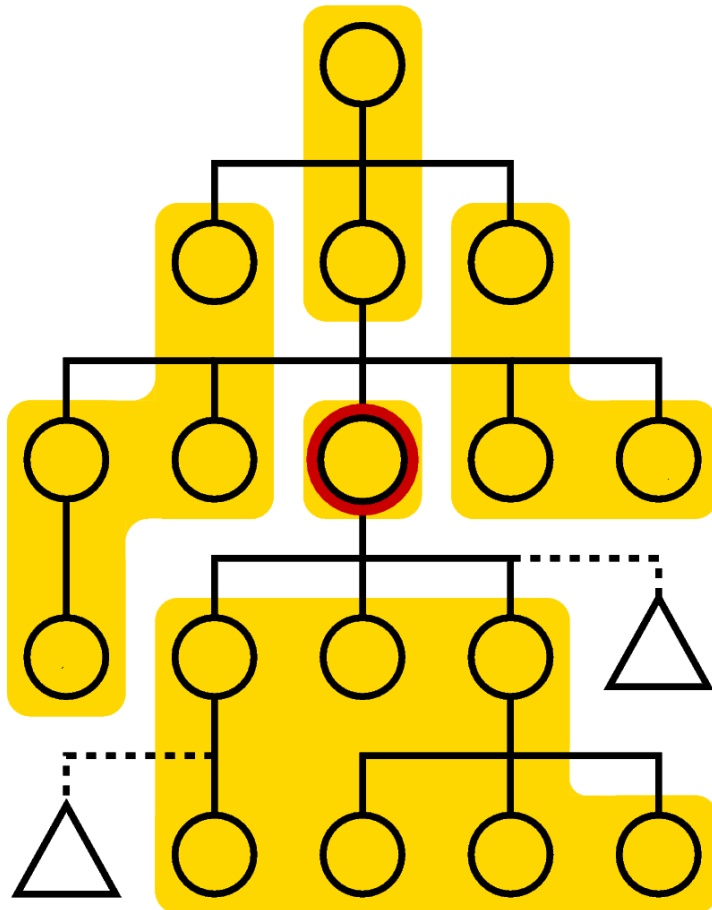
<!-- Ende -->
```

## XPath Achsen



- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- *following-sibling*-Achse
- *preceding*-Achse
- *following*-Achse

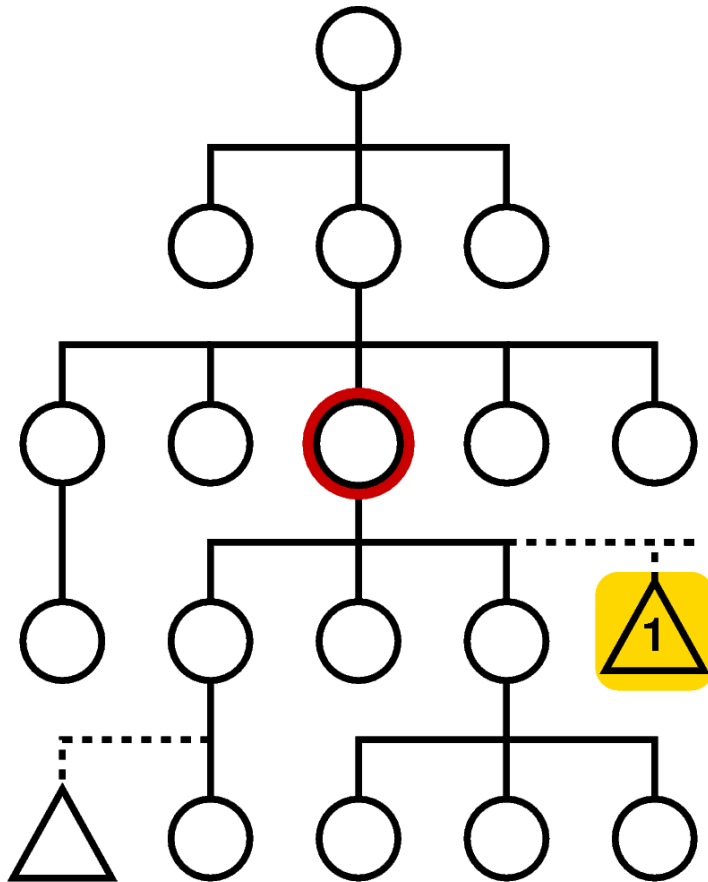
## XPath Achsen



- *parent*-Achse
- *self*-Achse
- *child*-Achse
- *ancestor*-Achse
- *ancestor-or-self*-Achse
- *descendant*-Achse
- *descendant-or-self*-Achse
- *preceding-sibling*-Achse
- *following-sibling*-Achse
- *preceding*-Achse
- *following*-Achse

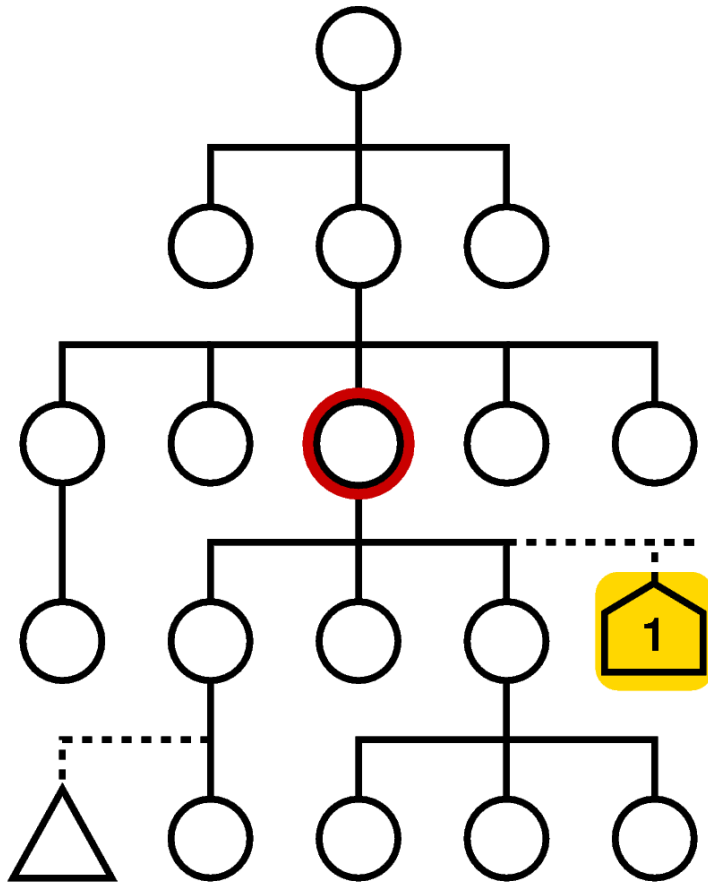
## XPath Achsen

- *attribute*-Achse
- *namespace*-Achse



## XPath Achsen

- *attribute*-Achse
- *namespace*-Achse



## XPath Knotentests

- Auswahl bestimmter Knoten aus der Achse
- String (Name der auszuwählenden Knoten)  
`following-sibling::chapter`
- Wildcard \* wählt alle Elementknoten aus  
`child::*`
- \* auch mit Namensraum-Präfixen  
`child::ns:*`
- @\* wählt alle Attribute aus  
`parent::@*`
- Mustersuche mit \* nicht möglich  
`child::art*`

## Spezielle Knotentests

- **node ( )**  
wählt alle Knoten aus
- **comment ( )**  
wählt alle Kommentarknoten aus
- **text ( )**  
wählt alle Textknoten aus
- **processing-instruction(Literal?)**  
wählt alle Verarbeitungsanweisungsknoten aus  
optional mit Namen als Argument

## XPath Prädikate

- Filterung der ausgewählten Knoten
- Syntax: **Prädikat ::= '[' Ausdruck ']'**  
**child::chapter[position() = 3]**
- Ausdruck wird zu *boolean* ausgewertet
- Falls Zahl: *true*, falls aktuelle Position  
**child::chapter[3]**
- Falls Knotenmenge: *true*, falls nicht leer  
**child::chapter[child::title]**

## XPath Kurzschreibweise

- `child::` kann weggelassen werden (default):  
`chapter`  $\equiv$  `child::chapter`
- `attribute::` kann mit `@` abgekürzt werden  
`preis/@waehrung`
- `self::node()` kann mit `.` abgekürzt werden
- `parent::node()` kann mit `..` abgekürzt werden  
`../title`
- `/descendant-or-self::node()/`  $\equiv$  `//`  
`./artikel`  
`//artikel`

## XPath Operatoren

- / und // für Pfadausdrücke
- | Vereinigung von Knotenmengen
- +, -, vor Minuszeichen muß Leerzeichen stehen
- \* Multiplikation nach Token außer @, ::, (, [, ,
- div Fließkommadivision
- mod Modulo
- or, and logische Operatoren
- =, !=, <=, <, >=, > Vergleichsoperatoren
- alle Operatoren sind linksassoziativ

## Funktionen für Knotenmengen

- **last()**  
Anzahl der Elemente in aktueller Knotenmenge
- **position()**  
aktuelle Position im Kontext
- **count(*node-set*)**

## Funktionen für Strings

- **string(*object?*)**  
wandelt Argument in String um
- **string-length(*string?*)**
- **normalize-space(*string?*)**

## Boolsche Funktionen

- **`boolean(object)`**  
wandelt Argument in booleschen Wert um
- **`not(boolean)`**
- **`true()`, `false()`**

## Numerische Funktionen

- **`number(object?)`**  
wandelt Argument in numerischen Wert um
- **`sum(node-set?)`**  
addiert Werte aller Knoten in gegebener Knotenmenge
- **`floor()`, `ceiling()`, `round()`**

## XPath Beispiele

- `not(ancestor::list)`
- `figure or table`
- `figure | table`
- `/doc/chapter[5]/section[2]`
- `//artikel[preis[@waehrung = 'EUR'] < 100]`
- `sum(//artikel/preis[@waehrung = 'EUR'])`

# Überblick

- **Einführung**  
Vorgeschichte, Entstehung der Sprachen
- **XPath**  
XML Lokalisierungssprache (XPath 1.0)
- **XQuery**  
XML Anfragesprache (XQuery 1.0)
- **Use Cases zu XQuery**
- **Ausblick**  
Zukunft von XQuery, Praxiseinsatz

## XQuery: Anwendungsgebiete

- Anfragen auf Dokumente  
z.B. Handbücher, Patientenakte
- Anfragen auf XML-Daten  
(physikalisch oder virtuell)
- Filterung von Datenströmen  
z.B. Logs, Nachrichten
- Suche in Katalogen von Dokument-Servern
- Verwendung in verschiedensten Umgebungen  
z.B. URL, XML, Code, Kommandozeile

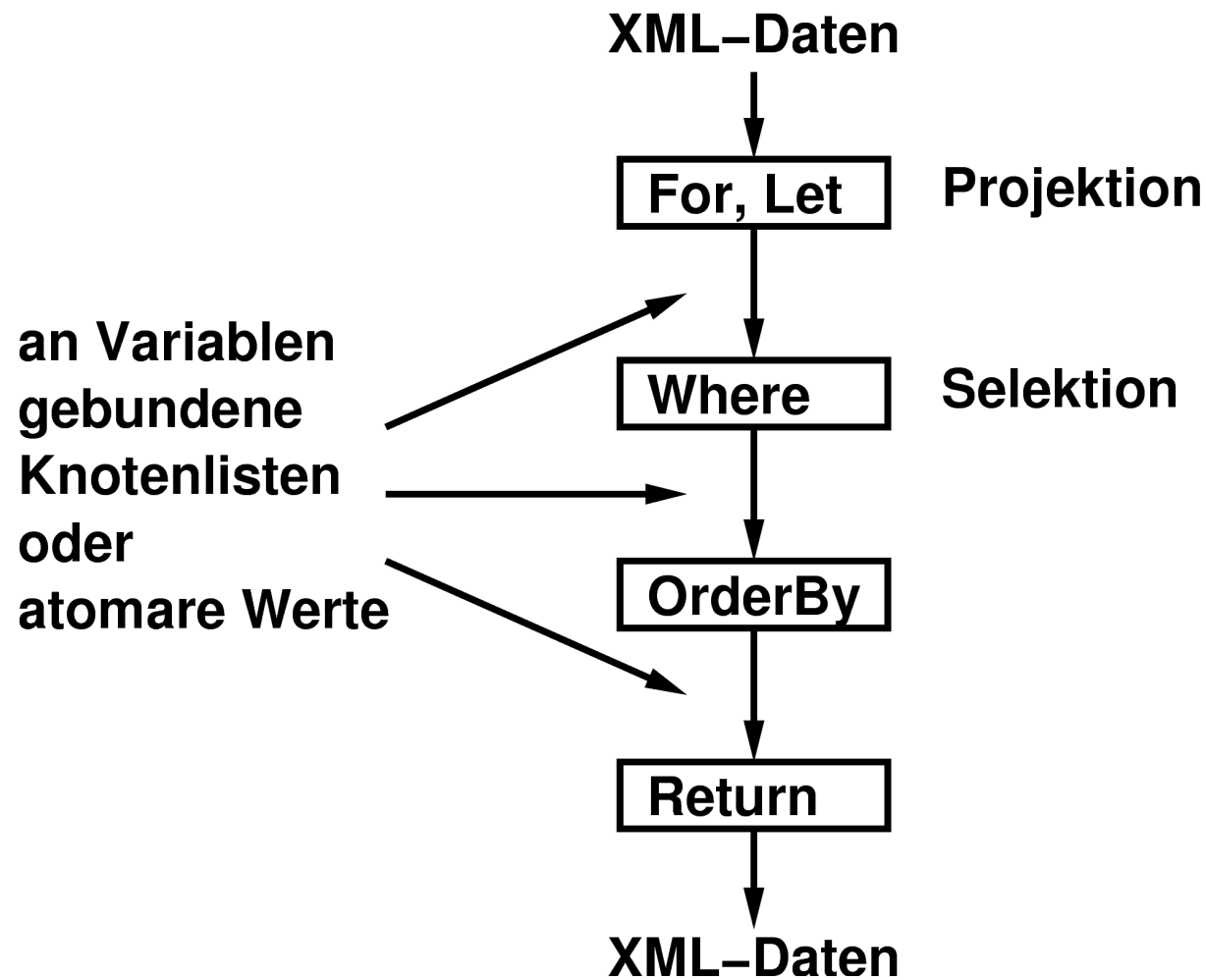
## XQuery: Anforderungen

- Lesbarkeit für Mensch und Maschine
- Deklarativität
- Protokollunabhängigkeit
- Unabhängigkeit von Datenhaltung
- Aufbau auf Informationen des Infoset
- Erweiterbarkeit
- Behandlung von NULL-Werten
- Aggregation

## XQuery 1.0 und XPath 2.0

- Zusammenarbeit der XQuery und XSL Arbeitsgruppen
- **XQuery 1.0 = XPath 2.0**
  - + **strenge Typisierung**
  - + **benutzerdefinierte Funktionen**
  - + **zusätzliche Ausdrücke**
- For-Let-Where-OrderBy-Return (FLWOR)
- XML Konstruktoren

## FLWOR-Ausdrücke: Informationsfluß



## XQuery: FLWOR-Ausdrücke

Beispiel:

```
for $d in document("depts.xml")//deptno
let $e := document("emp.xml")//emp[@dept = $d]
where count($e) >= 10
order by avg($e/salary) descending
return
  <result>
  {
    $d,
    <avgsal>{avg($e/salary)}</avgsal>
  }
</result>
```

## XQuery: FLWOR-Ausdrücke

**FLWOR ::= (For | Let)+ Where? OrderBy? Return**

- **For**: Iteration über eine Knotenmenge

**For ::= 'for' Var 'in' Expr  
( ',' Var 'in' Expr )\***

- **Let**: *Bindung* einer Knotenmenge an eine Variable  
Wert der Variable kann nachträglich nicht geändert werden

**Let ::= 'let' Var ':=' Expr  
( ',' Var ':=' Expr )\***

- **Where**: Einschränkung (analog zum WHERE in SQL)

**Where ::= 'where' Expr**

**Var ::= '\$' VarName TypeDeclaration?**

## XQuery: FLWOR-Ausdrücke

**FLWOR ::= (For | Let)+ Where? OrderBy? Return**

- **OrderBy**: Sortierung (analog zum ORDER BY in SQL)

**OrderBy ::= 'stable'? 'order' 'by' Expr Mod  
( ',' Expr Mod )\***

**Mod ::= ('ascending' | 'descending')?  
( 'empty' 'greatest' |  
'empty' 'least' )?  
( 'collation' StringLiteral )?**

- **Return**: Konstruktion des Ergebnisbaumes

**Return ::= 'return' Expr**

- beliebig verschachtelbar

## XQuery: Unterschied zwischen Let und For

```
let $s := (<one/>, <two/>, <three/>)  
return <out> { $s } </out>
```

## XQuery: Unterschied zwischen Let und For

```
let $s := (<one/>, <two/>, <three/>)  
return <out> { $s } </out>
```

```
<out>  
  <one/>  
  <two/>  
  <three/>  
</out>
```

## XQuery: Unterschied zwischen Let und For

```
for $s in (<one/>, <two/>, <three/>)  
return <out> { $s } </out>
```

## XQuery: Unterschied zwischen Let und For

```
for $s in (<one/>, <two/>, <three/>)  
return <out> { $s } </out>
```

```
<out>  
  <one/>  
</out>  
<out>  
  <two/>  
</out>  
<out>  
  <three/>  
</out>
```

## XQuery Konstruktoren

- Direktes Einsetzen (*statisch*):  

```
<preis waehrung="EUR">  
  179.00  
</preis>
```
- Einbetten von Ausdrücken (*dynamisch*):  

```
<preis waehrung="{ $art/price/curr }">  
  { $art/price/amount }  
</preis>
```
- auch Elementnamen können dynamisch erzeugt werden
- Listenkonstruktoren (`<one/>`, `<two/>`, `<three/>`)

## Neuigkeiten in XPath 2.0

- Listen statt Mengen
- Indexbereiche  
`chapter[1 to 5]`
- Vergleichsoperatoren `eq`, `ne`, `lt`, `gt`, `is`, `isnot`, `<<`, `>>`
- Listenoperatoren `union`, `intersect`, `except`
- Konditional  
`'if' Expr 'then' Expr 'else' Expr`
- Quantifizierte Ausdrücke `some`, `every`  
`every $w in $dept/worker`  
`satisfies $w/salary >= 40000`

## XQuery Datentypen und Funktionen

- strenges Typsystem, basierend auf XML Schema
- primitive Datentypen von XML Schema
- zusätzliche und abgeleitete Typen, eigene Typen
- Funktionen und Operatoren für alle Datentypen
- Aggregatfunktionen:  
**fn:count, fn:avg, fn:min, fn:max, fn:sum**
- Kontextfunktionen:  
**fn:context-item, fn:position, fn:current-date**
- Benutzerdefinierte Funktionen

## XQueryX

- alternative, "maschinenfreundliche" XML-Syntax
- kann leicht maschinell ausgewertet, umgeformt und generiert werden
- Beispiel:

```
let $authors := /book/author
return <authors> { $authors } </authors>
```

## XQueryX

```
<q:query
  xmlns:q="http://www.w3.org/2001/06/xqueryx">
  <q:flwr>
    <q:letAssignment variable="$authors">
      <q:step axis="CHILD">
        <q:identifier/>
        <q:step axis="CHILD">
          <q:identifier>book</q:identifier>
          <q:identifier>author</q:identifier>
        </q:step>
      </q:step>
    </q:letAssignment>
```

## XQueryX

```
<q:return>
  <q:elementConstructor>
    <q:tagName>
      <q:identifier>authors</q:identifier>
    </q:tagName>
    <q:variable>$authors</q:variable>
  </q:elementConstructor>
</q:return>
</q:flwr>
</q:query>
```

# Überblick

- **Einführung**  
Vorgeschichte, Entstehung der Sprachen
- **XPath**  
XML Lokalisierungssprache (XPath 1.0)
- **XQuery**  
XML Anfragesprache (XQuery 1.0)
- **Use Cases zu XQuery**
- **Ausblick**  
Zukunft von XQuery, Praxiseinsatz

# Überblick

- **Einführung**  
Vorgeschichte, Entstehung der Sprachen
- **XPath**  
XML Lokalisierungssprache (XPath 1.0)
- **XQuery**  
XML Anfragesprache (XQuery 1.0)
- **Use Cases zu XQuery**
- **Ausblick**  
Zukunft von XQuery, Praxiseinsatz

## Ausblick

- XML-Query-Arbeitsgruppe sehr groß
- Zahl der Implementierungen steigt
- noch nicht praxistauglich
- einige theoretische Probleme noch ungelöst
- nur lesender Zugriff
- IBM XTABLES
- Tamino