
Navigace v XML datech

Obsah

XPath	1
XPath - hlavní principy	1
XPath - aplikační oblasti	1
XPath - aplikační oblasti	2
XPath - aplikační oblasti	2
XPath - pojem cesty (paths) a lokace (locations)	2
XPath - syntaktická pravidla	2
XPath - osy (axes)	3
Příklad osa child	3
Příklad osa descendant	4
Příklad osa parent	4
Příklad osa ancestor	4
Příklad osa following-sibling	5
Příklad osa preceding-sibling	5
Příklad osa following	5
Příklad osa preceding	6
XPath - predikáty (predicates)	6
XPath - výrazy	6
XPath - zkrácená notace - Příklady	6
XPath - zkrácená notace (2)	7
Informační zdroje k XPath	7
XPath 2.0	7
XPath 2.0 - příklady	8

XPath

XPath - hlavní principy

- XPath je syntaxe pro specifikaci *částí* XML dokumentů (uzly, množiny uzlů, sekvence uzlů; nelze specifikovat *části* textových uzlů).
- XPath používá syntaxi obdobnou jako *cesty* v *souborovém systému*.
- XPath používá knihovnu standardních funkcí (evt. uživatelsky definovaných - v XPath 2.0 nebo i XPath 1.x, ale proprietárně - podle procesorů)
- XPath je od v 1.0 základem pro XSLT, od 2.0 i pro XQuery
- XPath syntaxe *není XML* (bylo by příliš "upovídáné")
- XPath 1.0 i 2.0 jsou doporučeními W3C (W3C Recommendation) - <http://www.w3.org/TR/xpath>

XPath - aplikační oblasti

- Pokročilá navigace v XML datech

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

```
<a>
  <b/>
  <b>
    <c/>
  </b>
  <b>
    <c/>
  </b>
</a>
```

- Vybrat třetí uzel b:

```
//b[3]
```

- Vybrat uzel b, který má potomka c:

```
//b[./c]
```

- Vybrat prázdný uzel b:

```
//b[count(./*)=0]
```

Příklad 1. //b[count(./*)=0]

XPath - aplikační oblasti

- Transformace (XSLT [<http://www.w3.org/TR/xslt/>])
 - slouží k výběru uzlů, které se mají zpracovávat

Příklad 2. <xsl:value-of select="./c"/>

XPath - aplikační oblasti

- V "selekční části" XML dotazovacích jazyků (XQuery [<http://www.w3.org/XML/Query/>])
- V některých modelovacích jazycích (Schematron [<http://www.schematron.com/>], XML Schema [<http://www.w3.org/XML/Schema/>])
- ...

XPath - pojem cesty (paths) a lokace (locations)

Cesta (path) určuje (tj. „naviguje nás na“) lokaci v dokumentu. Cesty jsou konstruovány podobně jako cesty v systému souborů, tj. jako

relativní vyhodnocovány vůči kontextovému uzlu (KU), viz dále, nebo

absolutní od kořene, ale výrazy (predikáty) také vyhodnocovány vůči KU

XPath - syntaktická pravidla

```
[20] PathExpr ::= AbsolutePathExpr | RelativePathExpr
```

```
[22] AbsolutePathExpr ::= ("/" RelativePathExpr?) | ("//" RelativePathExpr)
```

```
[23] RelativePathExpr ::= StepExpr ("/" | "//") StepExpr*
[24] StepExpr ::= AxisStep | GeneralStep
[25] AxisStep ::= (Axis? NodeTest StepQualifiers) | AbbreviatedStep
```

XPath - osy (axes)

Osy (jedn. číslo *axis*, množné *axes*) jsou množiny prvků dokumentu, vymezené (obvykle relativně) vůči *kontextu*.

Kontext je tvořen především *dokumentem* a *aktuálním (kontextovým) uzlem* (KU).

Osami jsou:

child	obsahuje dceřinné uzly kontextového (aktuálního) uzlu
descendant	obsahuje všechny potomky kontextového (aktuálního) uzlu (dále jen KU). Nepočítají se mezi ně atributy!!!
parent	obsahuje rodičovský uzel KU (existuje-li)
ancestor	obsahuje všechny předky - rodiče, "prarodiče", atd. až kořenový element (pokud KU není sám kořenový)
following-sibling	obsahuje všechny následující sourozence KU (pro NS a atributy je tato osa prázdná)
preceding-sibling	dtto, ale obsahuje <i>předchozí</i> sourozence
following	obsahuje všechny uzly nacházející se <i>po</i> KU (mimo atributů, potomků a NS uzlů)
preceding	dtto, ale obsahuje předchozí uzly (ale mimo předky, atributy, NS!)
attribute	obsahuje atributy (jen pro uzly - elementy)
namespace	obsahuje všechny NS uzly KU (jen pro uzly - elementy)
self	obsahuje samotný KU
descendant-or-self	obsahuje sjednocení os descendant a self
ancestor-or-self	obsahuje sjednocení os ancestor a self

Příklad osa child

Obrázek 1. //b/child::*

```
<?xml version="1.0"?>
<a>
  <b/>
  <b>
    <c/>
  </b>
  <b>
    <c/>
  </b>
```


Příklad osa descendant

Příklad 3. //b/descendant::*

```
<?xml version="1.0"?>
<a>
  <b/>
  <b>
    <c>
      <d/>
    </c>
  </b>
  <b>
    <c/>
  </b>
</a>
```

Příklad osa parent

Příklad 4. //d/parent::*

```
<?xml version="1.0"?>
<a>
  <b/>
  <b>
    <c>
      <d/>
    </c>
  </b>
  <b>
    <c/>
  </b>
</a>
```

Příklad osa ancestor

Příklad 5. //d/ancestor::*

```
<?xml version="1.0"?>
<a>
  <b/>
  <b>
    <c>
      <d/>
    </c>
  </b>
  <b>
    <c/>
  </b>
</a>
```

Příklad osa following-sibling

Příklad 6. //b/following-sibling::*

```
<?xml version="1.0"?>
<a>
  <b/>
  <b>
    <c>
      <d/>
    </c>
  </b>
  <b>
    <c/>
  </b>
</a>
```

Příklad osa preceding-sibling

Příklad 7. //b/preceding-sibling::*

```
<?xml version="1.0"?>
<a>
  <b/>
  <b>
    <c>
      <d/>
    </c>
  </b>
  <b>
    <c/>
  </b>
</a>
```

Příklad osa following

Příklad 8. /a/b/c/following::*

```
<?xml version="1.0"?>
<a>
  <b/>
  <b>
    <c>
      <d/>
    </c>
  <e/>
</b>
  <b>
    <c/>
  </b>
</a>
```

Příklad osa preceding

Příklad 9. /a/b/e/preceding::*

```
<?xml version="1.0"?>
<a>
  <b/>
  <b>
    <c>
      <d/>
    </c>
  </b>
  <b>
    <d/>
  </b>
</a>
```

XPath - predikáty (predicates)

Určeny k selekci (výběru) z uzlů specifikovaných např. cestou

př.: /article/para[3] - vybere třetí odstavec v článku

Nejjednodušším výrazem v predikátu je specifikace *pozice (blízkosti)* (proximity position) - viz výše

- Pozor u reverzních os (ancestor, preceding...) - pozice se počítá v rámci množiny uzlů vždy OD KONTEXTOVÉHO UZLU, tj. proti směru fyzického umístění v textové podobě dokumentu
- Specifikaci pozice **3** možno nahradit výrazem `position()=3`

XPath - výrazy

Určeny k použití v predikátech, k výpočtům, atd. Mohou obsahovat XPath funkce.

Výrazy mohou být:

- řetězcové
- numerické (hodnotami jsou floating-point čísla)
- logické (boolean)
- uzly
- sekvence

XPath - zkrácená notace - Příklady

- `para` vybere všechny dceřinné elementy kontextového uzlu jmenující se `para`
- `*` selects all element children of the context node
- `text()` selects all text node children of the context node

- @name *selects the name attribute of the context node*
- @* *selects all the attributes of the context node*
- para[1] *selects the first para child of the context node*
- para[last()] *selects the last para child of the context node*
- */para *selects all para grandchildren of the context node*
- /doc/chapter[5]/section[2] *selects the second section of the fifth chapter of the doc*
- chapter//para *vybere všechny element para, jež jsou následníky chapter*
- //para *vybere všechny elementy para z dokumentu*
- //olist/item *vybere všechny elementy item, které mají za rodiče olist. vybere kontextový uzel*
- ../para *vybere všechny elementy-potomky kontextového uzlu, které nesou značku para*
- .. *vybere rodičovský uzel od kontextového*
- ../@lang *vybere atribut lang rodičovského uzlu od kontextového*

XPath - zkrácená notace (2)

Nejpoužívanější zkracování je *osy child* :

- tj. píšeme `article/para` místo `child::article/child::para`.
- a `atributu: píšeme para[@type="warning"]` místo `child::para[attribute::type="warning"]`
- Další používané zkracování je `//` místo `/descendant-or-self::node()`
- a samozřejmě zkratky `.` a `..`



Poznámka

Pro přehlednost někdy delší formu zachováváme: nebraňme se jí za každou cenu!

Informační zdroje k XPath

- XPath na W3C: <http://www.w3.org/TR/xpath>
- Zvon XPath Tutorial: <http://zvon.org/xxl/XPathTutorial/Output/index.html>
- XPath Tutorial na W3Schools: http://www.w3schools.com/xpath/xpath_intro.asp

XPath 2.0

- Již finální specifikace - <http://www.w3.org/TR/xpath20/>
- Změna pohledu na hodnoty vrácené XPath výrazem: vše jsou **sekvence** (byť jednoprvkové)
- ->odstraňuje problémy s "pořadím" uzlů v množině

- Zavádí **podmíněné výrazy a cykly**
- Zavádí možnost uživatelských funkcí (psaných jako dynamicky vyhodnocované výrazy v XPath)
- Lze použít existenční a obecné kvantifikátory, např. `exist student/name="Fred"` nebo `all student/@id`
- Dále viz např. <http://www.saxonica.com/>, kde nalezneme i XPath/XSLT/XQuery procesor *Saxon*.

XPath 2.0 - příklady

- Řetězcové funkce [<http://www.fi.muni.cz/~tomp/xml03/xpath20/string.html>]
- Numerické funkce [<http://www.fi.muni.cz/~tomp/xml03/xpath20/numeric.html>]
- Funkce nad sekvencemi [<http://www.fi.muni.cz/~tomp/xml03/xpath20/sequence.html>]
- Booleovské funkce [<http://www.fi.muni.cz/~tomp/xml03/xpath20/boolean.html>]