

XMLコンソーシアムセミナー

XQuery技術概説

2006年10月4日

インフォテリア株式会社
教育部

Copyright © 2006 Infoteria Corporation All rights reserved.

Table of Contents

- XQueryとは
- XQueryの特徴
- 関連技術、仕様書群
- FLWOR式
- for節
- let節
- for節による結合
- where節
- order by節とreturn節
- その他の主な機能

Copyright © 2006 Infoteria Corporation All rights reserved.

XQueryとは

XMLに対する問い合わせ言語

W3Cで標準化

現在、Candidate Recommendation

XPath式を拡張した言語

XQuery 1.0はXPath 2.0を拡張

(XPath 2.0はXPath 1.0のスーパーセット)

XQuery 1.0とXPath 2.0のデータモデルは共通

XQueryXも用意されている

XQueryXはXQuery構文のXML表現

クエリー自体をプログラム処理する際に有効

Copyright © 2006 Infoteria Corporation All rights reserved.

XQueryの特徴

XML名前空間仕様に準拠

名前空間仕様に従っていないXMLは処理できない

クエリー本文に先立って名前空間を宣言できる

ノードの集合には順序がある

ノードもしくは値の順序つき集合をシーケンスと呼ぶ

ノードもしくは値をアイテムと呼ぶ

つまりシーケンスは、0以上のアイテムの順序つき集合

関数型言語

変数の値は不変

式の入れ子が可能

型付けされた言語

型の定義はXML Schemaを使用する

型を問うことができる (instance of)

型をキャストすることができる (cast as)

Copyright © 2006 Infoteria Corporation All rights reserved.

関連技術、仕様書群

W3C Candidate Recommendation

XML Path Language (XPath) 2.0
XQuery 1.0: An XML Query Language
XQuery 1.0 and XPath 2.0 Formal Semantics
XQuery 1.0 and XPath 2.0 Data Model (XDM)
XQuery 1.0 and XPath 2.0 Functions and Operators
XML Syntax for XQuery 1.0 (XQueryX)
XSLT 2.0 and XQuery 1.0 Serialization

W3C Working Draft

XQuery Update Facility (XQueryを拡張するもの)
XQuery 1.0 and XPath 2.0 Full-Text (XQueryを拡張するもの)

Copyright © 2006 Infoteria Corporation All rights reserved.

FLWOR式

for, let, where, order by, returnキーワードによる基本構文
for ... シーケンスをバインドし、個々のアイテムを順に処理する
let ... シーケンスをバインドする (繰り返さない)
where ... for節、let節をフィルタリングする
order by ... タブルの並べ替え
return ... 結果のタブルの構築

Copyright © 2006 Infoteria Corporation All rights reserved.

for節

～ シーケンスをバインドし、個々のアイテムを順に処理する ～

```
XQuery
for $n in (<n>1</n>, <n>2</n>, <n>3</n>)
return <out>{$n}</out>
```

```
Result
<out>
  <n>1</n>
</out>
<out>
  <n>2</n>
</out>
<out>
  <n>3</n>
</out>
```

Copyright © 2006 Infoteria Corporation All rights reserved.

let節

～ シーケンスをバインドする (繰り返さない) ～

```
XQuery
let $n := (<n>1</n>, <n>2</n>, <n>3</n>)
return <out>{$n}</out>
```

```
Result
<out>
  <n>1</n>
  <n>2</n>
  <n>3</n>
</out>
```

Copyright © 2006 Infoteria Corporation All rights reserved.

for節による結合

XQuery	Result
for \$i in (<n>1</n>, <n>2</n>)	<out>
for \$j in (<n>2</n>, <n>3</n>)	<n>1</n><n>2</n>
return	</out>
<out>	<out>
{ \$i } { \$j }	<n>1</n><n>3</n>
</out>	</out>
	<out>
	<n>2</n><n>2</n>
	</out>
	<out>
	<n>2</n><n>3</n>
	</out>

Copyright © 2006 Infotera Corporation All rights reserved.

where節

~ for節、let節をフィルタリングする ~

XQuery	Result
for \$i in (<n>1</n>, <n>2</n>)	<out>
for \$j in (<n>2</n>, <n>3</n>)	<n>1</n><n>2</n>
where \$i != \$j	</out>
return	<out>
<out>	<n>1</n><n>3</n>
{ \$i } { \$j }	</out>
</out>	<out>
	<n>2</n><n>3</n>
	</out>

Copyright © 2006 Infotera Corporation All rights reserved.

order by節とreturn節

～ タブルの並べ替え、結果のタブルの構築 ～

XQuery	Result
for \$i in (<n>1</n>, <n>2</n>)	<out>
for \$j in (<n>2</n>, <n>3</n>)	<n>2</n><n>3</n><sum>5</sum>
let \$sum := \$i + \$j	</out>
where \$i != \$j	<out>
order by \$sum descending	<n>1</n><n>3</n><sum>4</sum>
return	</out>
<out>	<n>1</n><n>2</n><sum>3</sum>
{ \$i } { \$j } <sum>{ \$sum }</sum>	</out>
</out>	</out>

Copyright © 2006 Infoteria Corporation All rights reserved.

その他の主な機能(1)

Module Declaration

～ 変数や関数の集合をモジュールとして定義できる

例) module namespace math = "http://example.org/math-functions";

Boundary-space Declaration

～ ダイレクト要素コンストラクタ内の空白の扱いを定める

例) declare boundary-space preserve;

Ordering Mode Declaration

～ シーケンスをorderedで扱うか、もしくはunorderedで扱うかを定める

例) declare ordering unordered;

Copyright © 2006 Infoteria Corporation All rights reserved.

その他の主な機能(2)

Schema Import

~ スキーマのインポート

例) `import schema namespace
soap="http://www.w3.org/2003/05/soap-envelope"
at "http://www.w3.org/2003/05/soap-envelope/";`

Module Import

~ モジュールのインポート

例) `import module namespace
math = "http://example.org/math-functions";`

その他の主な機能(3)

Namespace Declaration

~ 名前空間の宣言

例) `declare namespace foo = "http://example.org";
<foo:bar> Lentils </foo:bar>`

Variable Declaration

~ 変数の宣言、外部から値を与えることもできる

例) `declare variable $x as xs:integer := 7;
例) declare variable $x as xs:integer external;`

その他の主な機能(4)

Function Declaration

~ 独自の関数を定義できる

例) `declare function local:depth($e as node()) as xs:integer`
`{`
 (: A node with no children has depth 1 :)
 (: Otherwise, add 1 to max depth of children :)
 if (fn:empty(\$e/*)) then 1
 else fn:max(for \$c in \$e/* return local:depth(\$c)) + 1
`};`

`local:depth(fn:doc("partlist.xml"))`