

データベース入門 資料 6

XSLT の反復構造

平成 23 年 5 月 21 日

目 次

1	XSLT の反復構造	1
1.1	xsl:for-each 要素による反復	1
2	データの並べ換えと空白文字の出力	2
2.1	xsl:sort 要素による並べ換え	2
2.2	xsl:text 要素を使った空白文字の出力	3
3	練習問題	3

1 XSLT の反復構造

本章では、XSLT テンプレートを前回資料 2.1 の XML 文書例¹ (以下、単に「XML 文書例」と呼ぶ。) に適用するとして、XSLT の反復構造を説明する。

1.1 xsl:for-each 要素による反復

XSLT スタイルシートのテンプレート内に xsl:for-each 要素を置くことで、テンプレート内で反復処理を指定することができる。次に示すのは xsl:for-each を使ったスタイルシートの例である。

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text" encoding="UTF-8" />
  <xsl:template match="/">
    <xsl:for-each select="/ref/book">
      <xsl:value-of select="title" />
      <xsl:value-of select="author" />
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

これを XML 文書例に適用すると、次の出力が得られる²。

```
Title1Author1Title2Author2
```

以下では、上記のスタイルシートのうち、xsl:for-each の動作に関わる部分について説明する。

- <xsl:template match="/">

出力を具体的に記述する部分（テンプレート）の始まり。

属性 match="/" の指定により、ルートノード (/) がカレントノードになるため、テンプレート内の初期の相対ロケーションパスの基点は / になる。

- <xsl:for-each select="/ref/book">

xsl:for-each 要素は、select 属性で指定されたノードの集まり（ノードセット）の中の各ノードを逐次カレントノードにしながら、終了タグ </xsl:for-each> の前まで（要素の内容）に記述された処理を繰り返して行うものである。

XML 文書例を処理する場合、/ref/book は二つのノードに対応するので、xsl:for-each 要素の内容は 2 回反復処理される。

なお、この開始タグが現れた段階でのカレントノードは / であるため、select 属性値を相対ロケーションパスの ref/book や ./ref/book としても動作は同じ。

¹http://echoes.hak.hokkyodai.ac.jp/db/550/xslt.html#sec:xml_sample

²xsltproc /pub/db/xml/for_each.xsl /pub/db/xml/data_model.xml

- `<xsl:value-of select="title" />`
`<xsl:value-of select="author" />`

xsl:for-each 要素による繰り返しの対象部分。XML 文書例に対しては、次の動作が起きる。

反復 1 回目 第 1 の /ref/book 要素 (/ref/book[1]) がカレントノードとなる。

このとき、xsl:value-of の select 属性値である title と author (相対ロケーションパスでの指定) は、絶対ロケーションパスでの /ref/book[1]/title および /ref/book[1]/author に等しい。

従って、xsl:value-of 要素によって、それらの文字列値である Title1 と Author1 が出力される。

反復 2 回目 第 2 の /ref/book 要素 (/ref/book[2]) がカレントノードとなる。

反復 1 回目と同様に、xsl:value-of によって、/ref/book[2]/title および /ref/book[2]/author の文字列値である Title2 と Author2 が出力される。

- `</xsl:for-each>`

xsl:for-each 要素の終了タグ。xsl:for-each 要素が終わると、カレントノードは以前のもの (このスタイルシートでは xsl:template で指定された /) に戻る。そのため、この後に相対ロケーションパスを含むテンプレート記述があれば、その起点は再度 / になる。

2 データの並べ換えと空白文字の出力

第 1.1 節の XSLT スタイルシートにおいて、xsl:for-each 要素を

```
<xsl:for-each select="/ref/book">
  <xsl:sort order="descending"/>
  <xsl:value-of select="title" />
  <xsl:text> </xsl:text>
  <xsl:value-of select="author" />
  <xsl:text>
</xsl:text>
</xsl:for-each>
```

に変更して、XML 文書例を処理すると、出力は次のとおりとなる。

```
Title2 Author2
Title1 Author1
```

この出力を得るために行っているテンプレート内の処理について、以下に記す。

2.1 xsl:sort 要素による並べ換え

xsl:for-each 要素は、特に指定しなければ、XML 文書に現れた順にカレントノードを変更して反復処理を行うが、xsl:for-each の内容の始めに xsl:sort 要素を置くことで、処理の順序を変更できる。

例えば、先の `<xsl:sort order="descending"/>` は、出力される文字列値が辞書の逆順（降順）になるように、`xsl:for-each` が処理するノードの順序を並べ換えるものである。

`xsl:order` 要素の動作を変更する属性は幾つかあり、`order` 属性は並べ換えの順を指定するものである。属性値には `ascending`（昇順）か `descending`（降順）を指定する（`order` 属性を略すと昇順になる）。

他の属性としては、並べ換えに使うキーを指定する `select` 属性があり、`<xsl:sort select="author" order="descending"/>` とすれば、`author` ノードの文字列値が降順になるように、処理順序の並べ換えが行われる。

2.2 xsl:text 要素を使った空白文字の出力

第 1.1 節の実行結果において、空白や改行無しに `Title1` や `Author1` などが続いて出力されたのは、テンプレート内で、空白文字（ブランクや改行）からなる文字データをタグの外に書いても、それらが出力されないためである。

テンプレートに記述した空白や改行を出力したければ、`xsl:text` 要素が使える。例えば、内容が空白文字である `xsl:text` 要素

```
<xsl:text> </xsl:text>
```

や

```
<xsl:text>
</xsl:text>
```

をテンプレート内に書けば、空白や改行を出力できる。

3 練習問題

1. 第 1.1 節の XSLT スタイルシートにおける `xsl:value-of` 要素の `select` 属性値 `title` と `author` を、各々 `/ref/book/title` と `/ref/book/author` に変更して、XML 文書例 (`/pub/db/xml/data_model.xml`) を `xsltproc` コマンドで処理してみなさい。その出力が得られる理由を考えなさい。
2. 参考文献の XML データ (`/pub/db/xml/ref.xml`) から、`title` 要素の内容のみを全て出力するための XSLT テンプレートを作成し、`xsltproc` コマンドで処理しなさい。`xsl:for-each` 要素を使うこと。`xsl:text` 要素を使って、適切な改行を出力すること。
`xsl:sort` 要素の使い方も確認してみなさい。
3. 「資料 4. XML の基礎」の 4. 練習問題³ の 4. で作成した郵便番号の XML 文書から、同問題の設問にあるような郵便番号一覧を作るための XSLT テンプレートを作成しなさい。ただし、各列の標題である「市名 町名 郵便番号」の出力は省いてよい。

³http://echoes.hak.hokkyodai.ac.jp/db/550/xml_basic.html