

データベース入門 資料 5

XSLT スタイルシートとロケーションパス

平成 23 年 5 月 16 日

目 次

1	XSLT	1
1.1	XSLT	1
2	XSLT における XML 文書のデータモデル	1
2.1	木構造データモデル	1
2.2	ノードの指定方法 (ロケーションパス)	2
2.2.1	絶対ロケーションパス	3
2.2.2	ノードセット	3
2.2.3	ノードの絞り込み (filtering)	3
2.2.4	相対ロケーションパス	3
3	XSLT スタイルシートの基本構造	4
3.1	XSLT における相対ロケーションパスの基点	5
4	文字列値と <code>xsl:value-of</code> 要素の使い方	6
4.1	文字列値	6
4.2	<code>xsl:value-of</code> 要素の使い方	6
5	XSLT プロセッサ	6
6	練習問題	7

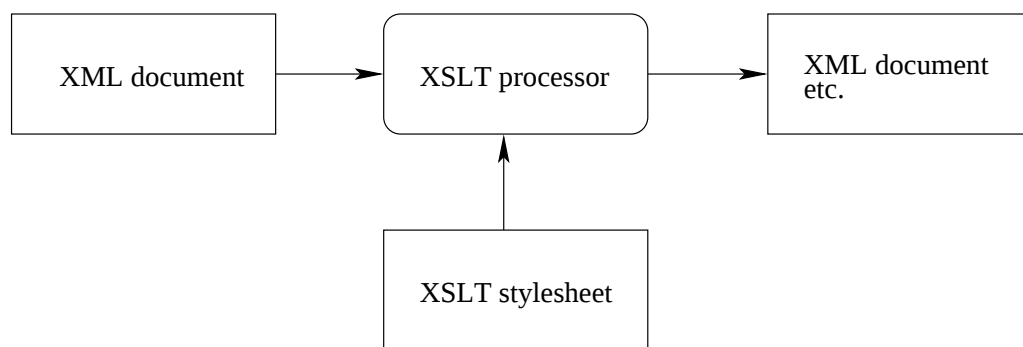
1 XSLT

この資料では XSLT 言語の基礎と、XML 文書からデータの検索や抽出を行うための方法を学ぶ。

1.1 XSLT

XSLT (XSL Transformation) ¹は XML 文書を変換・加工するための言語であり、XSLT を用いれば、XML 文書から必要なデータのみを取り出したり、それを Web ページ用の文書にする等のデータ操作ができる。

XML 文書をどのように変換するかを書いた XSLT スタイルシートを作り、XML 文書と XSLT スタイルシートを XSLT プロセッサで処理して結果を得る。



2 XSLT における XML 文書のデータモデル

2.1 木構造データモデル

XSLT を使うには、変換処理やデータ抽出の対象となる XML 文書内の箇所（範囲）を指定する必要がある。そのためには XSLT が XML 文書をどのように構造化して扱うか（データモデル）の理解が必要である。

図 2.1 の左側に XML 文書の一例を示す。この文書のルート要素は ref であり、ref の内容は文字データの Reference、および book や web 等の要素である。

XSLT では、XML 文書を木 (tree) 構造を持つデータとして扱う。図 2.1 の右側は、左側の XML 文書を XSLT のデータモデル (木) として描画したものである。この木における / や ref, #TEXT などをもノード (node) と呼ぶ²。

この木には、次に示す 4 種類のノードがある³。

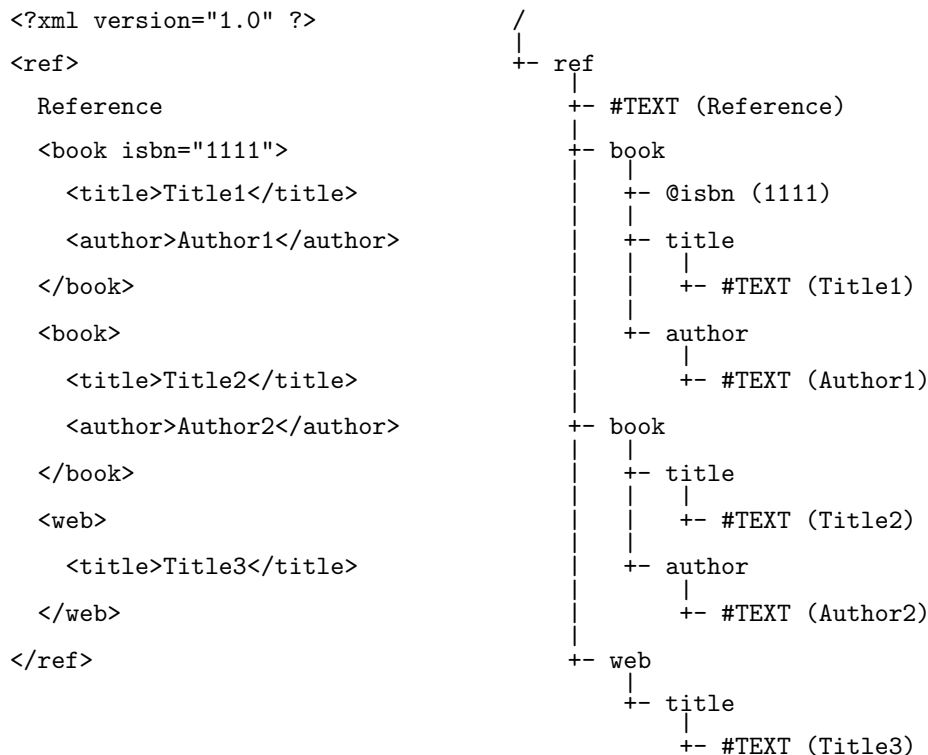
ルートノード (root node) XML 文書全体に対応するノードであり / と表記される。ルートノード以外のすべてのノードは、ルートノードの子孫である。

要素ノード (element node) XML 文書の各要素に対応するノード。XML 文書における「ルート要素」は、データモデルにおいてはルートノードの子である。個々の要素ノードの名前は XML 文書における要素名である。

¹XSL は eXtensible Stylesheet Language の略。

²木やノードはグラフ理論における用語であり、直感的には、ノード（節点）と節点同士をつなぐ辺 (edge) の集まりがグラフである。閉路を持たないグラフが木である。節点 (node) を頂点と、辺 (edge) を枝と呼ぶこともある。

³XSLT で使われるノードには、この他にコメントノード、処理命令ノード、名前空間ノードがある。



右側の木における表記の意味は以下のとおり。

- / はルートノード
- 小文字表記は要素ノード名または属性ノード名
ただし、属性ノード名には @ を付加した
- #TEXT はテキストノード
- 空白文字（ブランクや改行等）のみからなるテキストノードの記載は省略
- テキストノードと属性ノードには（ ）内に文字列値を記した

図 1: XSLT における XML 文書のデータモデル (ノード表現)

属性ノード (attribute node) XML 文書の属性に対応するノード。属性ノードは要素ノードの子になる。個々の属性ノードの名前は XML 文書における属性名である。

テキストノード (text node) XML 文書内の文字データに対応するノード。先の XML 文書では、< > の外にある Reference や Title1 等が文字データであり、これらに対応するのがテキストノードである。個々のテキストノードに名前は無い。なお、テキストノードの文字データは親要素ノードが持つ文字列値として得られるため、テキストノードを意識する場面はあまり無い。

2.2 ノードの指定方法 (ロケーションパス)

XML 文書をモデル化した木からノードを指定する記述がロケーションパス (location path) である。この資料で扱うのはロケーションパスの省略記法のみであり、この場合、ロケーションパスの基本的な記述法はディレクトリやファイルのパス名と同じである。

なお、ロケーションパスは XPath (XML Path Language) という規格で定められている⁴。

⁴XML 関係の仕様はすべて <http://www.w3.org/> から閲覧可能。記述は英語だが、日本語訳が存在する文書もある。

2.2.1 絶対ロケーションパス

絶対ロケーションパス (absolute location path) は、ルートノードを基点としてノードの位置を表現するものである。絶対ロケーションパスでは、ルートノードを表す `/` に続き、ルートノードから当該ノードに至る道筋にある下記のノードを `/` で区切って書き並べる⁵。

要素ノード 要素ノードを含んだロケーションパスでは、要素ノード名がそのまま使える。

例：`/ref/web/title`

属性ノード ロケーションパスで属性ノードを指定するには、名前の前に `@` を付ける。

例：`/ref/book/@isbn`

2.2.2 ノードセット

ファイルやディレクトリのパス名では、パス名に対応するファイルやディレクトリがただ一つであるのに対し、XML のロケーションパスは一般にノードの集まり (ノードセット; node-set) に対応する。例えば、第 2.1 節の木において、`/ref/book` に対応するのは二つのノードからなるノードセットである。

ノードセットの指定には、次のようなロケーションパスを用いることもできる。

<code>/ref/*</code>	ref ノードのすべての子要素ノード
<code>/ref/book/@*</code>	book ノードのすべての子属性ノード
<code>//title</code>	ルートノードの子孫であるすべての title 要素ノード
<code>/ref//title</code>	ref ノードの子孫の中のすべての title 要素ノード
<code>/ref/book /ref/web</code>	ref ノードの子要素ノード book と web をあわせたノードセット

2.2.3 ノードの絞り込み (filtering)

次の例のように、ロケーションパスに `[...]` を記述することで、ノードセットから ... の条件を満たすノード (またはノードセット) のみを指定したことになる⁶。

<code>/ref/book[2]</code>	2 番目の book 要素ノード
<code>/ref/book[@isbn='1111']</code>	isbn 属性が 1111 である book 要素ノード
<code>/ref/book[author]</code>	author 要素ノードを子に持つ book 要素ノード
<code>/ref/book[2]/title</code>	2 番目の book ノードの子である title 要素ノード

2.2.4 相対ロケーションパス

ルートノード以外を基点とするロケーションパスを相対ロケーションパス (relative location path) という。相対ロケーションパスでは、ディレクトリやファイルの相対パス名と同様に、ロケーションパスの始めに `/` を書かない。

相対ロケーションパスでは、基点となるノードを `.` (ドット) で表し、親ノードを `..` (ドット二つ) で表す。

⁵テキストノードには名前がないので、テキストノードのロケーションパスには `text()` を用いる。例：`/ref/text()`

⁶`[ ]` の部分をロケーションパスの述部という

第 2.1 節の木において、仮りに web を基点とすれば、相対ロケーションパスの例は次のとおりとなる。

```
web の子ノードの title は ./title または title
ref は .. であり, / は ../..
isbn は ../book/@isbn
```

相対ロケーションパスの実際の基点は、XSLT では、カレントノード (current node) と呼ばれる、現在処理中のノードである。(= ドット . で表されるノード) となる⁷。カレントノードについては、3.1 で説明する。

なお、相対ロケーションパスでも、第 2.2.3 節で紹介したノードの絞り込みが可能である。

3 XSLT スタイルシートの基本構造

XSLT スタイルシートは、XML 文書をどのような形に変換するかを、XSLT で記したものである⁸。なお、以下では XSLT スタイルシートを、単にスタイルシートと呼ぶことがある。

次の例は、変換対象となる XML 文書から、XML 宣言やタグ等を除いた文字データ部分をすべて出力するスタイルシートである。

```
<?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:value-of select="/" />
  </xsl:template>
</xsl:stylesheet>
```

XSLT スタイルシート自体も整形形式の XML 文書なので、スタイルシートにも前回資料にある XML の用語 (XML 宣言や要素、属性等) や規則が当てはまる。加えて、XSLT 独自の規則等がある。上記スタイルシートにおける、各記述の意味や役割を以下に記す。

- <?xml version="1.0" encoding="UTF-8" ?> : XML 宣言
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

XSLT スタイルシートには xsl:stylesheet 要素が必要である。この要素の内容 (この次の行から終了タグ </xsl:stylesheet> の前まで) は、XSLT の規則に従って記述しなければならず、ここに記述される xsl: で始まる名前の要素には、XSLT に固有の意味や役割がある。

xsl:stylesheet 要素の子要素を、XSLT の用語でトップレベル要素 (top-level element) という。トップレベル要素として記述できる要素は限定されており、この資料では xsl:output と xsl:template 要素のみを用いる。

⁷相対ロケーションパスの基点は一般には文脈ノード (context node) と呼ばれる。カレントノードは XSLT の用語であり、文脈ノードは XPath での用語。

⁸Web ページ作成に用いられる CSS (Cascading Style Sheets) とは異なる

- `<xsl:output method="text" encoding="UTF-8" />`

xsl:output 要素は、変換結果の出力形式を定める。ここでは出力形式を text 形式⁹、出力の文字コードを UTF-8 としている。

- `<xsl:template match="/">`

xsl:template 要素は、変換元の XML 文書の中の

1. どの部分に関して、
2. 何を出力するか、

を具体的に記述する、スタイルシートの中核をなす要素である。

1. を指定するのが xsl:template 要素の match 属性であり、属性値 / は変換元の XML 文書全体を指している。

2. は xsl:template 要素の内容として記述する。この記述をテンプレートという。

テンプレートには、XSLT により意味が定められている要素や、その他の出力したい文字データ等を書いていく。

テンプレート内に記述した文字データは、基本的にはそのまま出力されるが、空白文字（blank や改行）のみからなる文字データは出力されない。出力の細かな振る舞いは xsl:output 要素の method 属性の指定によって変わる。

なお、一つのスタイルシート内に match 属性値の異なる xsl:template 要素を複数記述することが可能であるが、扱いが煩雑となるため、この資料で扱うスタイルシートでは、属性 match="/" の xsl:template 要素を一つだけ記述することとする。

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

xsl:value-of 要素を用いると、変換対象となる XML 文書から、select 属性で指定したものを取り出すことができる。

この要素の使い方は後述するが、select="/" を指定すれば、xsl:value-of 要素は XML 文書全体から文字データ部分をすべて取り出す。

これを xsl:template 要素の内容として記述することで、XML 文書の文字データ部分をすべて出力するスタイルシートを作成したことになる。

3.1 XSLT における相対ロケーションパスの基点

XSLT では、カレントノード (current node) と呼ばれる、現在処理中のノードが相対ロケーションパスの基点 (= ドット . で表されるノード) である。

XSLT スタイルシートでは、ロケーションパスをテンプレート内で使うが、その際の (初期の) カレントノードは xsl:template 要素の match 属性によって決まる。例えば、`<xsl:template match="/">` で始まるテンプレート内の初期カレントノードはルートノードである。従って、このテンプレートでは / と . は等価であり、`<xsl:value-of select="/" />` と `<xsl:value-of select="." />` の出力は同じである。

⁹text 形式は変換結果からタグを除去 (テキストノードの内容のみ出力) し、文字データのエスケープ (変換) を行わないで出力する形式。text の他には xml と html がある。

テンプレート内にカレントノードを変更する XSLT の要素があれば、その中ではカレントノードに応じて相対ロケーションパスの基点が変わる¹⁰。

4 文字列値と `xsl:value-of` 要素の使い方

4.1 文字列値

文字列値 (string-value) は各ノードの値であり、`xsl:value-of` 要素を使う際に必要である。文字列値は、ノードの種類毎に、次のとおりに定められている。

テキストノード: XML 文書内の当該文字データ

属性ノード: 属性値

ルートノード、要素ノード: 当該ノードの子孫となるテキストノードの文字列値を出現順に連結したもの

これは、要素ノードに対応する XML 文書の要素から、タグ `<...>` を除いたものに相当する

4.2 `xsl:value-of` 要素の使い方

`xsl:value-of` 要素は、XSLT スタイルシート内の `xsl:template` 要素 (テンプレート) 内で使える XSLT の要素であり、書式は次の通りである。

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

expression として、単一のノードに対応するロケーションパスを指定すれば、`xsl:value-of` はそのノードの文字列値を XML 文書から取り出す。複数のノード (ノードセット) に対応するロケーションパスを指定すれば、ノードセット中の最初のノードの文字列値のみを取り出す。

expression には、次の例のように、XSLT の関数も書ける。

```
<xsl:value-of select="count(/ref/book)" />
```

関数 `count` は、引数に与えられたノードセットの個数を返すものであり、この `xsl:value-of` 要素を含むスタイルシートを第 2.1 節の XML 文書に適用すれば、`/ref/book` のノード数である 2 が出力される。

5 XSLT プロセッサ

XSLT を使うには、XSLT プロセッサと呼ばれるプログラムが必要である。これは単独のコマンドとして用意されていることや、多くの Web ブラウザのようにアプリケーションに組み込まれている場合がある。

ここでは、XSLT プロセッサとして `xsltproc` コマンドを用いる。実行の書式は次のとおり。

```
xsltproc [option] stylesheet file ...
```

¹⁰`xsl:for-each` 要素が該当する。これは次回に扱う。

ここで、*stylesheet* と *file* は、各々、XSLT スタイルシートと XML 文書の入ったファイルの名前である。例えば、

```
xsltproc /pub/db/xml/all.xsl /pub/db/xml/ref.xml
```

は、XSLT スタイルシート /pub/db/xml/all.xsl の内容に則って XML 文書 /pub/db/xml/ref.xml を変換し、結果を標準出力（画面）に出力する。結果をファイルに入れたければ、リダイレクト (> *file*) を使えばよい。

6 練習問題

以下の問題では、xsltproc コマンドを使って、指定された XML 文書に XSLT スタイルシートを適用し、処理しなさい。今回の練習問題では、出力における空白や改行の乱れは無視してよい。

1. <xsl:template match="/">を開始タグとするテンプレート内に、

```
/ref/web/title, /ref/book/@isbn, /ref/book, /ref/book[2], /ref/book[@isbn='1111'],  
/ref/book[author], /ref/book[2]/title, ./ref/web/title
```

を select 属性値とする xsl:value-of 要素を記述して（一つのスタイルシートに追加してよい）、第 2.1 節の XML 文書 (/pub/db/xml/data_model.xml) を処理し、各出力が得られる理由を確認しなさい。

2. 任意の XML 文書から title 要素の総数を出力する XSLT スタイルシートを作成し、第 2.1 節の XML 文書および前回資料の XML 文書 (/pub/db/xml/ref.xml) に適用しなさい。
3. 前回資料 (4. XML の基礎) の練習問題 4. で作成した XML 文書から、自宅の郵便番号を出力する XSLT スタイルシートを作成しなさい。ただし、スタイルシートには、自宅の町名のロケーションパスを記述すること。