

# 情報科学演習 参考資料

## XSLT による Web ページ文書の生成

平成 30 年 6 月 11 日

### 目次

<b>1</b>	<b>XML 文書から Web ページ用文書を作る</b>	<b>1</b>
1.1	データと表現形式 . . . . .	1
1.2	HTML . . . . .	1
1.3	XML から HTML への変換 . . . . .	2
1.4	Web ブラウザを使った XSLT 変換 — XML 文書への処理命令の追加 . . . . .	4
<b>2</b>	<b>練習問題</b>	<b>5</b>

# 1 XML 文書から Web ページ用文書を作る

これまでの XSLT の利用目的は、主に XML 文書から必要なデータを抽出することであった。しかしながら、XSLT は XML 文書を適切な表現形式に変換する目的一般に使用されるものである。その一例として、この章では XML 文書を利用した Web ページ用文書作成の基本的な方法を紹介する。

## 1.1 データと表現形式

Web ページはデータを人に提示するのに適した表現形式の一つであり、XML 文書から Web ページ用の文書を作ることができれば、人がデータを閲覧するには便利である。

一方で、例えば、文書を用紙に印刷して閲覧したいこともある。もちろん Web ページをそのまま印刷して使うことも可能であるものの、Web ページはコンピュータの画面等で読むことを前提として作られているため、そのまま印刷すると用紙上の行数や文字数等が不適切になったり、ハイパーリンクが使えなくなるために読みにくい文書となることがある。かといって、Web と印刷用に、同じデータを別々に用意するのは適切ではない。複数の文書用にデータを用意しておく、データを更新する際などに、各文書用のデータをそれぞれ更新する手間が生じる上に、文書間でデータの不整合が起きる可能性がある。データは一つの XML 文書として作成しておき、データの表現形式を定める役割を XSLT テンプレートに負わせれば、元の XML 文書だけを更新して、予め作成しておいた Web ページ作成用 XSLT テンプレートと印刷文書作成用のテンプレートを適用するだけで文書の更新が済む。

このように、XSLT による XML の変換を使えば、データを格納したひとつの XML 文書を様々な形で表現することが可能になる。

## 1.2 HTML

HTML (HyperText Markup Language) は Web ページ記述用のマークアップ言語である。XML と HTML は共に SGML (Standard Generalized Markup Language) というマークアップ言語を基に開発されたものなので、両者の記法や用語は似ている。XML との違いは、使えるタグの名前が予め決められている点や、HTML のバージョンによっては開始タグのみの要素があることやタグ名に大文字・小文字の区別がないこと等である。なお、HTML はあくまで Web ページ (人間が読むもの) の記述用言語であり、XML のような汎用性のあるデータ記述言語ではない。

以下に簡単な HTML の例を示す。ここでは、body 要素の中のタグ (h1, h2, a, p, table) の使い方が分かれば十分である (Web ブラウザによる出力は [こちら](#))。

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8" />
    <title>Sample HTML</title>
  </head>
  <body>
    <h1>HTML 文書のサンプル</h1>
```

```

<h2>段落</h2>
<p>p タグは段落を作ります。</p>
<h2>ハイパーリンク</h2>
<p>外部文書へのリンクには href 属性を持つ a タグを使います
  (例:<a href="http://echoes.hak.hokkyodai.ac.jp/db/30/">授業のページへの
リンク</a>)。
  同じコンピュータ内のファイルにリンクを張るときには,
  href 属性値にファイルのパス名が使えます。
  同じディレクトリ内のファイルであれば, href="ファイル名" でいいです。
</p>
<h2>表</h2>
<p>表を作るには table タグを使います。
  表の中に行を作るのは tr タグです。
  行の中に何か書くには td タグを使います。行や列や表題は th タグです。
</p>
<table>
  <tr>
    <th></th><th>列 A</th><th>列 B</th>
  </tr>
  <tr>
    <th>行 1</th><td>1A</td><td>1B</td>
  </tr>
  <tr>
    <th>行 2</th><td>1A</td><td>2B</td>
  </tr>
</table>
</body>
</html>

```

サーバ上の Web ページではなく、自分で作成したファイルを Web ブラウザで直接に表示するには、ブラウザのメニュー等からファイルを開く操作を行うか、URL 欄に `file://` に続きファイルの絶対パス名 (例: `file:///pub/eis/xml/sample.html`) を入力する等をすればよい。w3m というページャーでは、コマンドラインの引数に `.html` で終わるファイル名を指定して実行すると (例: `w3m /pub/eis/xml/sample.html`)、w3m は自動的にそのファイルを HTML 文書として解釈して表示する。

### 1.3 XML から HTML への変換

XSLT を使って XML 文書を HTML 文書に変換すれば、XML 文書内のデータを Web ページとして見るができる。

以下に、XSLT スタイルシートの例 (`/pub/eis/xml/ref2html.xsl`) を示す。これは**参考文献の入った XML 文書** (`/pub/eis/xml/ref.xml`) を元に HTML 文書を出力するスタイルシートである。これまでに示した XSLT テンプレートと異なるのは、次の 3 点である。

1. `xsl:output` 要素の `method` 属性値を `html` にすることで、テンプレートに記述したタグを出力させる<sup>1</sup>。
2. テンプレートに XSLT の要素以外のもの (HTML のタグ) を記述している。
3. HTML の `href` 属性値を XML 文書から得るために、属性値テンプレート `{location-path}` を用いている。

XSLT スタイルシートのテンプレート記述において、タグの属性値部分に XML 文書のデータを埋め込みたければ、`{ }` 内にロケーションパスを記述すればよい。これを属性値テンプレート (attribute value template) という。

下記のスタイルシートでは、`<a href="{uri}">` において、XML 文書内の `uri` 要素の文字列値を取得し、それを Web ページのリンク先 URI (URL) として使う。

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8" />
<xsl:template match="/">
<html>
  <head>
    <title>Reference List</title>
  </head>
  <body>
    <h1>Reference List</h1>
    <h2>book</h2>
    <table>
<xsl:for-each select="ref/book">
  <tr>
    <td><xsl:value-of select="@isbn" /></td>
    <td><xsl:value-of select="title" /></td>
    <td><xsl:value-of select="author|editor" /></td>
  </tr>
</xsl:for-each>
</table>
    <h2>web</h2>
    <table>
<xsl:for-each select="ref/web">
  <tr>
    <td><a href="{uri}"><xsl:value-of select="title" /></a></td>
  </tr>
</xsl:for-each>
```

<sup>1</sup>これまで利用してきた `<xsl:output method="text">` は、`xsl:template` の内容のうち、タグ等は出力せずに文字データのみを出力する指定である。

```

    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>

```

参考文献の XML 文書にこのスタイルシートを適用して HTML 文書を出力するには、

```
xsltproc /pub/eis/xml/ref2html.xsl /pub/eis/xml/ref.xml
```

とすればよい。その HTML 文書を w3m を使って Web ページとして閲覧したければ、リダイレクトを使って出力を一旦ファイルに保存してから読み込むか、パイプを使って

```
xsltproc /pub/eis/xml/ref2html.xsl /pub/eis/xml/ref.xml | w3m -T text/html
```

でよい<sup>2</sup>。

また、Firefox 等の Web ブラウザが持っている XSLT プロセッサの機能を使って XML 文書を変換し、Web ページとして表示することもできる ([http://echoes.hak.hokkyodai.ac.jp/ref\\_PI.xml](http://echoes.hak.hokkyodai.ac.jp/ref_PI.xml))。その方法は第 1.4 節で述べる。

## 1.4 Web ブラウザを使った XSLT 変換 — XML 文書への処理命令の追加

XML パーサを持った Web ブラウザは XML 文書の構造を解析して表示できる。さらに XSLT プロセッサの機能も併せ持つ Web ブラウザを使えば、XSLT による XML 文書の変換と結果の表示を Web ブラウザのみで行える。

そのためには、変換対象となる XML 文書内に、適用したいスタイルシート of 情報を書き込む必要がある。これを XML では処理命令 (PI; processing instruction) という。処理命令を記述する位置は、ルート要素よりも前としなければならない<sup>3</sup>。

次のものは、XSLT による変換を行う処理命令の例である。適用したい XSLT スタイルシートは href 属性で指定する。Web サイト上にスタイルシートがある場合には、

```

<?xml-stylesheet type="text/xsl"
  href="http://echoes.hak.hokkyodai.ac.jp/ref2html.xsl" ?>

```

のような URL によるスタイルシートの指定が可能であり、XML 文書と同じコンピュータにあるファイルをスタイルシートとして指定するのであれば

```
<?xml-stylesheet type="text/xsl" href="/pub/eis/xml/ref2html.xsl" ?>
```

のようにファイルのパス名を記述すればよい<sup>4</sup>。

処理命令が書かれた XML 文書を Web ブラウザで開けば、XSLT 変換の結果を得ることができる<sup>5</sup>。

<sup>2</sup>w3m のオプションは man w3m を実行して調べることができる。

<sup>3</sup>もちろん XML 宣言より後ろ。

<sup>4</sup>XML 文書と同じディレクトリに存在する XSLT テンプレートを適用する場合、href 属性には、パス名に代え、ファイル名のみ記述してよい。さらに、絶対パス名の前に file:// というスキームを付けると、ローカルに存在するファイルであることを明記した URL を記述したことになる。

<sup>5</sup>実際に処理命令を実行できるかどうかは、XML 文書を扱うプログラム (Web ブラウザ) に依存する。

## 2 練習問題

1. 資料 7 で利用した XML 文書 (/pub/eis/xml/data\_model.xml) に、第 1.3 節の XSLT スタイルシート例 (/pub/eis/xml/ref2html.xsl) を適用し、結果を出力しなさい。xsltproc コマンドを使うこと。

その出力をパイプと less を使って確認してから、リダイレクトを使ってファイルに保存しなさい。ファイル名の終り（拡張子）は .html とする。保存したファイルを Web ブラウザで表示しなさい<sup>6</sup>。

2. 「8. XSLT スタイルシートの基本構造と反復処理」の練習問題 3. で作成した XSLT スタイルシートを、HTML を出力するように変更しなさい。

また、XML 文書に処理命令を追加することにより、xsltproc コマンドを使う代わりに、ブラウザの XSLT プロセッサ機能を使って表を閲覧できるようにしなさい。

---

<sup>6</sup>w3m を利用すればよいが、firefox を使うのであればローカル (Windows) で X サーバを起動して X を使える状態にしてから、リモートで firefox -no-remote を実行すれば、firefox をリモートのコンピュータで起動して、ウィンドウをローカルに表示できる。